Opencdp restful接口

目录

[Opencdp restful接口 1](#_Toc493668497)

[Restful接口基本操作 4](#_Toc493668498)

[列出所有restful接口 5](#_Toc493668499)

[登陆 6](#_Toc493668500)

[用户帐号 6](#_Toc493668501)

[list所有用户 6](#_Toc493668502)

[用户详情 8](#_Toc493668503)

[创建用户 8](#_Toc493668504)

[删除用户 9](#_Toc493668505)

[更新用户 9](#_Toc493668506)

[更改密码 9](#_Toc493668507)

[修改用户权限 9](#_Toc493668508)

[存储主机管理 11](#_Toc493668509)

[主机列表 11](#_Toc493668510)

[创建其他主机 12](#_Toc493668511)

[主机端initiator 12](#_Toc493668512)

[列表 12](#_Toc493668513)

[创建 13](#_Toc493668514)

[列出主机端所有光纤节点 13](#_Toc493668515)

[列出所有主机端光纤targets节点 13](#_Toc493668516)

[设置一个光纤端口为target模式 14](#_Toc493668517)

[取消一个光纤target端口 14](#_Toc493668518)

[主机端远程clone设置 15](#_Toc493668519)

[列表 15](#_Toc493668520)

[创建远程主机clone设置 16](#_Toc493668521)

[主机网卡列表 16](#_Toc493668522)

[网络设置 17](#_Toc493668523)

[硬盘 17](#_Toc493668524)

[Cdp设备可使用外接存储作为自己的存储空间 17](#_Toc493668525)

[存储池管理 29](#_Toc493668526)

[列表 29](#_Toc493668527)

[创建存储池 31](#_Toc493668528)

[在存储池中创建块设备卷 31](#_Toc493668529)

[在存储池中创建文件卷 31](#_Toc493668530)

[往池中增加硬盘设备 33](#_Toc493668531)

[从池中删除块设备 34](#_Toc493668532)

[卷管理 34](#_Toc493668533)

[列表 34](#_Toc493668534)

[列出卷的快照 35](#_Toc493668535)

[创建卷的快照 35](#_Toc493668536)

[删除卷的快照 35](#_Toc493668537)

[快照卷的clone 35](#_Toc493668538)

[客户端管理 35](#_Toc493668539)

[列表列出所有已注册客户端 35](#_Toc493668540)

[列出所有可用客户端包括未注册的 37](#_Toc493668541)

[注册一个客户端 38](#_Toc493668542)

[判断连接是否online 38](#_Toc493668543)

[通过一个连接获取相关agent的操作系统相关信息 38](#_Toc493668544)

[通过连接取到的值存入agent的oldgrains字段 47](#_Toc493668545)

[通过连接获取系统的磁盘情况 47](#_Toc493668546)

[客户端的initiator管理 51](#_Toc493668547)

[映射 52](#_Toc493668548)

[客户端脚本插件管理 53](#_Toc493668549)

[日志设备 55](#_Toc493668550)

[列表： 55](#_Toc493668551)

[给块设备包开启日志设备 56](#_Toc493668552)

[察看日志 56](#_Toc493668553)

[给日志设备创建一个标志 56](#_Toc493668554)

[在一个块设备上回放指定日志记录 57](#_Toc493668555)

[通过此id可查询此任务的详细执行情况 57](#_Toc493668556)

[RAID卷管理 57](#_Toc493668557)

[RAID1的列表 58](#_Toc493668558)

[MirrorVG创建卷组镜像 59](#_Toc493668559)

[删除MirrorVG卷 60](#_Toc493668560)

[RAID1创建 60](#_Toc493668561)

[删除RAID卷 63](#_Toc493668562)

[镜像卷的管理 63](#_Toc493668563)

[镜像的列表 63](#_Toc493668564)

[创建镜像设置 64](#_Toc493668565)

[创建agent -> agent的mirror设置 66](#_Toc493668566)

[修改agentendpoint端信息 66](#_Toc493668567)

[将mirror设置里的配置安装drbd设备，执行后将分别在agent端和本地端执行创建drbd设备 67](#_Toc493668568)

[开启agent-〉local初始化进程 69](#_Toc493668569)

[开启同步到本地方向，agent数据将开始同步到本地盘上 70](#_Toc493668570)

[开启同步到远程，将本地盘数据同步到agent的盘上 71](#_Toc493668571)

[将指定agent设置为primary 71](#_Toc493668572)

[将agent设置为secondary 71](#_Toc493668573)

[查询同步完成的进度 71](#_Toc493668574)

[卸载drbd设备，但不删除mirror配置 72](#_Toc493668575)

[只安装agent端的drbd设备 72](#_Toc493668576)

[查询agent端角色 73](#_Toc493668577)

[Upagent端卷 73](#_Toc493668578)

[Downagent端卷 73](#_Toc493668579)

[暂停同步 73](#_Toc493668580)

[重新开启同步 74](#_Toc493668581)

[删除mirror配置 74](#_Toc493668582)

[文件同步节点管理 74](#_Toc493668583)

[所有同步节点列出 74](#_Toc493668584)

[增加同步节点 75](#_Toc493668585)

[删除同步节点 75](#_Toc493668586)

[修改 75](#_Toc493668587)

[文件克隆 75](#_Toc493668588)

[列出文件克隆设置 76](#_Toc493668589)

[增加文件克隆设置 76](#_Toc493668590)

[修改文件克隆设置 77](#_Toc493668591)

[删除文件克隆设置 77](#_Toc493668592)

[启动文件克隆从agent到local 77](#_Toc493668593)

[启动文件克隆从local到agent 78](#_Toc493668594)

[samba共享设置 78](#_Toc493668595)

[列表 78](#_Toc493668596)

[增加网络共享 79](#_Toc493668597)

[删除网络共享 79](#_Toc493668598)

[NFS网络共享 79](#_Toc493668599)

[列表 79](#_Toc493668600)

[创建nfs共享 80](#_Toc493668601)

[修改nfs共享 81](#_Toc493668602)

[删除nfs共享 81](#_Toc493668603)

[物理盘复制任务 81](#_Toc493668604)

[长时间任务结果及调度 83](#_Toc493668605)

[任务结果列表 83](#_Toc493668606)

[根据id查找任务详情 88](#_Toc493668607)

[创建调度任务周期调度 89](#_Toc493668608)

[创建调度任务crontab调度 89](#_Toc493668609)

[修改调度任务的参数 90](#_Toc493668610)

[修改调度任务的调度时间 90](#_Toc493668611)

[启动调度任务 90](#_Toc493668612)

[停止调度任务 91](#_Toc493668613)

[同时支持task,和status组合查找 91](#_Toc493668614)

[删除调度任务 91](#_Toc493668615)

[自动创建快照任务 91](#_Toc493668616)

[自动文件克隆任务 92](#_Toc493668617)

[虚拟云管理 92](#_Toc493668618)

[列出云主机配置 92](#_Toc493668619)

[创建云主机连接配置 93](#_Toc493668620)

[创建虚拟机 93](#_Toc493668621)

[集群的管理 93](#_Toc493668622)

[集群的列表 94](#_Toc493668623)

[Cephpool管理 99](#_Toc493668624)

[集群pool的列表 99](#_Toc493668625)

[修改cephpool 100](#_Toc493668626)

[删除cephpool 100](#_Toc493668627)

[Cephclient管理 100](#_Toc493668628)

[Cephclient列表 101](#_Toc493668629)

[Cephclient增加 101](#_Toc493668630)

[Cephosd 列表 102](#_Toc493668631)

[Ceph osd增加 103](#_Toc493668632)

[删除指定OSD 103](#_Toc493668633)

[其他节点更新ceph 信息 104](#_Toc493668634)

[系统服务 104](#_Toc493668635)

[获取所有服务 104](#_Toc493668636)

[设置某个服务 106](#_Toc493668637)

[审计系统 106](#_Toc493668638)

[察看审计信息 106](#_Toc493668639)

[查询审计可以根据时间,返回值组合查找 110](#_Toc493668640)

[删除审计日志 112](#_Toc493668641)

[全部删除 112](#_Toc493668642)

[单条记录删除 112](#_Toc493668643)

[支持组合条件删除 112](#_Toc493668644)

# Restful接口基本操作

RESTFUL接口使用http协议，基本操作GET,POST,PUT,DELETE,对应语义为查询，增加，修改，删除

返回数据为json格式，默认支持分页，默认返回分页大小为20，

如果需要修改分页大小使用参数limit

http://192.168.40.23:8000/api/volumes?limit=10

{

"count": 12,

"next": "http://192.168.40.23:8000/api/volumes?limit=10&offset=10",

"previous": null,

"results": [

{

"url": "http://192.168.40.23:8000/api/volumes/1",

"id": 1,

"name": "Virtual\_disk pci-0000\_00\_10\_0-scsi-0\_0\_0\_0",

"uuid": "7f7072d0-4379-4202-94b1-1b55f41c9e0e",

"createdate": "2016-03-22T09:07:08.820431Z",

…

# 列出所有restful接口

<http://192.168.40.23:8000/api/>

GET

{

"users": "<http://192.168.40.23:8000/api/users>",

"contenttypes": "<http://192.168.40.23:8000/api/contenttypes>",

"hosts": "<http://192.168.40.23:8000/api/hosts>",

"hostgroups": "<http://192.168.40.23:8000/api/hostgroups>",

"netdevices": "<http://192.168.40.23:8000/api/netdevices>",

"ipaddresses": "<http://192.168.40.23:8000/api/ipaddresses>",

"peers": "<http://192.168.40.23:8000/api/peers>",

"cmdlogs": "<http://192.168.40.23:8000/api/cmdlogs>",

"disks": "<http://192.168.40.23:8000/api/disks>",

"pools": "<http://192.168.40.23:8000/api/pools>",

"volumes": "<http://192.168.40.23:8000/api/volumes>",

"snapshots": "<http://192.168.40.23:8000/api/snapshots>",

"agents": "<http://192.168.40.23:8000/api/agents>",

"agentgroups": "<http://192.168.40.23:8000/api/agentgroups>",

"connects": "<http://192.168.40.23:8000/api/connects>",

"luns": "<http://192.168.40.23:8000/api/luns>",

"initiators": "<http://192.168.40.23:8000/api/initiators>",

"portals": "<http://192.168.40.23:8000/api/portals>",

"logdev": "<http://192.168.40.23:8000/api/logdev>",

"results": "<http://192.168.40.23:8000/api/results>",

"schedules": "<http://192.168.40.23:8000/api/schedules>",

"mirrors": "<http://192.168.40.23:8000/api/mirrors>"

}

# 登陆

1. 使用http Auth机制

用户名密码认证

curl <http://192.168.40.23:8000/api/users> -u zktx:Aa123456

1. 使用Token方式

Token 97cc2cfb4ede1ddb72a625cababf485913722bc9

curl http://192.168.40.23:8000/api/users -H "Authorization: Token 97cc2cfb4ede1ddb72a625cababf485913722bc9"

# 用户帐号

## list所有用户

<http://192.168.40.23:8000/api/users>

GET

返回例子

**HTTP 200 OK**

**Allow:** GET, POST, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/users/1>",

"id": 1,

"username": "zktx",

"email": "[zktx@heartsone.com](mailto:zktx@heartsone.com)",

"first\_name": "wang",

"last\_name": "mingxin",

"is\_active": true,

"is\_staff": true,

"is\_superuser": true,

"last\_login": "2016-06-15T06:10:49.685718Z",

"date\_joined": "2016-03-07T10:40:30.568919Z",

"token": {

"token": "97cc2cfb4ede1ddb72a625cababf485913722bc9",

"createdate": "2016-03-17T07:01:50.765Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/users/2>",

"id": 2,

"username": "admin",

"email": "admin@test",

"first\_name": "administrator",

"last\_name": "for opencdp",

"is\_active": true,

"is\_staff": true,

"is\_superuser": true,

"last\_login": null,

"date\_joined": "2016-06-15T08:56:51.725026Z",

"token": {

"token": "Not set yet!"

}

}

]

}

## 用户详情

<http://192.168.40.23:8000/api/users/1>

GET

{

"url": "<http://192.168.40.23:8000/api/users/1>",

"id": 1,

"username": "zktx",

"email": "[zktx@heartsone.com](mailto:zktx@heartsone.com)",

"first\_name": "wang",

"last\_name": "mingxin",

"is\_active": true,

"is\_staff": true,

"is\_superuser": true,

"last\_login": "2016-06-15T06:10:49.685718Z",

"date\_joined": "2016-03-07T10:40:30.568919Z",

"token": {

"token": "97cc2cfb4ede1ddb72a625cababf485913722bc9",

"createdate": "2016-03-17T07:01:50.765Z"

}

}

## 创建用户

<http://192.168.40.23:8000/api/users>

POST

{

"username": "admin",

“password”:”admin”,

"email": "[zktx@heartsone.com](mailto:zktx@heartsone.com)",

"first\_name": "wang",

"last\_name": "mingxin",

"is\_active": true,

"is\_staff": true,

"is\_superuser": true

}

## 删除用户

<http://192.168.40.23:8000/api/users/2>

DELETE

## 更新用户

<http://192.168.40.23:8000/api/users/2>

PUT

{

}

## 更改密码

<http://192.168.40.23:8000/api/users/2/chgpasswd>

PUT

{

“password”:”123456”

}

## 修改用户权限

<http://192.168.50.23:8000/api/users/8/chgperm>

PUT

{

"permissions": [

"cmdlog.delete\_auditentry"

]

}

{

"url": "<http://192.168.50.23:8000/api/users/8>",

"id": 8,

"username": "test",

"email": "",

"first\_name": "",

"last\_name": "",

"is\_active": true,

"is\_staff": true,

"is\_superuser": false,

"last\_login": null,

"date\_joined": "2017-09-08T07:41:40Z",

"token": {

"token": "Not set yet!"

},

"permissions": [

"cmdlog.delete\_auditentry"

]

}

# 存储主机管理

## 主机列表

http://192.168.40.23:8000/api/hosts

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"id": 1,

"name": "localhost.localdomain",

"url": "<http://192.168.40.23:8000/api/hosts/1>",

"netdevice\_set": [

"<http://192.168.40.23:8000/api/netdevices/1>",

"<http://192.168.40.23:8000/api/netdevices/2>",

"<http://192.168.40.23:8000/api/netdevices/3>",

"<http://192.168.40.23:8000/api/netdevices/4>"

],

"hostgroup\_set": [],

"primary\_ip\_address": {

"id": 3,

"url": "<http://192.168.40.23:8000/api/ipaddresses/3>",

"device": "<http://192.168.40.23:8000/api/netdevices/2>",

"address": "192.168.40.23/24",

"gateway": "",

"nameservers": "",

"domain": "",

"configure": true,

"primary\_address": true

}

}

]

}

## 创建其他主机

<http://192.168.40.23:8000/api/hosts>

POST

{

"name": "centos-local",

"hosturl": "http://192.168.40.23:8000/api",

"token": "72aff6e664b9f8f7dc322766f7aad0615e50a2ae"

}

## 主机端initiator

### 列表

<http://192.168.40.23:8000/api/hostinitiators>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/hostinitiators/1>",

"id": 1,

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"wwn": "12:34:56:78:90",

"type": "qla2xxx"

}

]

}

### 创建

<http://192.168.40.23:8000/api/hostinitiators>

POST

{

"wwn": "12:34:56:78:90",

"host":1,

"type": “qla2xxx”

}

host:主机id

type: “iscsi” or “qla2xxx” initiator类型

如果创建的是光纤模式,将自动把处于target模式的光纤节点取消掉

### 列出主机端所有光纤节点

<http://192.168.50.172/api/api/hosts/getfcports>

GET

{

"wwns": [

"21:00:00:24:ff:46:c0:c4",

"21:00:00:24:ff:46:c0:c5"

]

}

### 列出所有主机端光纤targets节点

<http://192.168.40.23:8000/api/hosts/getfctargets>

GET

返回所有target模式的wwns

{

"wwns": [

"21:00:00:24:ff:46:c0:c4",

"21:00:00:24:ff:46:c0:c5"

]

}

### 设置一个光纤端口为target模式

<http://192.168.40.23:8000/api/hosts/enable_fctarget>

POST

{

“wwn”: “21:00:00:24:ff:46:c0:c4”

}

返回所有target模式的wwn号

{

"wwns": [

"21:00:00:24:ff:46:c0:c4",

"21:00:00:24:ff:46:c0:c5"

]

}

### 取消一个光纤target端口

<http://192.168.40.23:8000/api/hosts/disable_fctarget>

POST

{

“wwn”: “21:00:00:24:ff:46:c0:c4”

}

返回target模式的wwn号

{

"wwns": [

"21:00:00:24:ff:46:c0:c5",

]

}

## 主机端远程clone设置

### 列表

<http://192.168.40.23:8000/api/remotemirrors>

GET

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"id": 1,

"url": "<http://192.168.40.23:8000/api/remotemirrors/1>",

"srcHost": "<http://192.168.40.23:8000/api/hosts/1>",

"dstHost": "<http://192.168.40.23:8000/api/hosts/4>",

"srcvolid": 43,

"dstvolid": 8822,

"dstport": 9201,

"task": "<http://192.168.40.23:8000/api/results/20216>",

"options": null

},

{

"id": 2,

"url": "<http://192.168.40.23:8000/api/remotemirrors/2>",

"srcHost": "<http://192.168.40.23:8000/api/hosts/4>",

"dstHost": "<http://192.168.40.23:8000/api/hosts/1>",

"srcvolid": 43,

"dstvolid": 8822,

"dstport": 9201,

"task": "<http://192.168.40.23:8000/api/results/20200>",

"options": null

}

]

### 创建远程主机clone设置

<http://192.168.40.23:8000/api/remotemirrors>

POST

{

"srcHost": 4,

"dstHost": 1,

"srcvolid": 43,

"dstvolid": 8822,

"dstport": 9201,

"options": '{"gzip": false, "max\_snap": 200}'

}

srcHost 源主机id

dstHost 目的主机id

srcvolid 源卷id

dstvolid 目的卷id

dstport 网络传输用的目的端端口号

options ：传输选项json格式字符串，gzip 是否压缩，max\_snap 最大保留快照数

## 主机网卡列表

### 网络设置

### 硬盘

Cdp设备可使用外接存储作为自己的存储空间

#### 外接iscsi类型的存储

外接iscsi存储

##### 列表

<http://192.168.40.23:8000/api/iscsiportals>

GET

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"id": 1,

"url": "<http://192.168.40.23:8000/api/iscsiportals/1>",

"address": "192.168.50.172",

"port": 3260

},

{

"id": 2,

"url": "<http://192.168.40.23:8000/api/iscsiportals/2>",

"address": "192.168.40.74",

"port": 3260

}

]

}

##### 创建外接iscsi存储

<http://192.168.40.23:8000/api/iscsiportals>

POST

{

“address”:”192.168.40.74”,

“port”:3260

}

##### 扫描iscsiportal上的所有target

<http://192.168.50.23:8000/api/iscsiportals/1/discovery>

GET

[

[

"192.168.50.172:3260,1",

"iqn.2003-01.org.linux-iscsi.localhost.x8664:wang"

],

[

"192.168.50.172:3260,1",

"iqn.2003-01.org.linux-iscsi.localhost.x8664:win4059"

],

[

"192.168.50.172:3260,1",

"iqn.2003-01.org.linux-iscsi.localhost.x8664:70122sys"

]

]

将返回通过此portal扫描到的全部target，并将新的target存入数据库

##### 列出所有iscsi发现的target

<http://192.168.50.23:8000/api/iscsitargets>

GET

{

"count": 4,

"next": null,

"previous": null,

"results": [

{

"id": 1,

"url": "<http://192.168.50.23:8000/api/iscsitargets/1>",

"portalid": "192.168.50.172:3260,1",

"target": "iqn.2003-01.org.linux-iscsi.localhost.x8664:wang",

"iscsiportal": "<http://192.168.50.23:8000/api/iscsiportals/1>"

},

{

"id": 2,

"url": "<http://192.168.50.23:8000/api/iscsitargets/2>",

"portalid": "192.168.50.172:3260,1",

"target": "iqn.2003-01.org.linux-iscsi.localhost.x8664:juan4083",

"iscsiportal": "<http://192.168.50.23:8000/api/iscsiportals/1>"

},

{

"id": 3,

"url": "<http://192.168.50.23:8000/api/iscsitargets/3>",

"portalid": "192.168.50.172:3260,1",

"target": "iqn.2003-01.org.linux-iscsi.localhost.x8664:win4059",

"iscsiportal": "<http://192.168.50.23:8000/api/iscsiportals/1>"

},

{

"id": 4,

"url": "<http://192.168.50.23:8000/api/iscsitargets/4>",

"portalid": "192.168.50.172:3260,1",

"target": "iqn.2003-01.org.linux-iscsi.localhost.x8664:70122sys",

"iscsiportal": "<http://192.168.50.23:8000/api/iscsiportals/1>"

}

]

}

##### 列出所有已经login的target

<http://192.168.50.23:8000/api/iscsiportals/listsession>

GET

[

[

"192.168.50.172:3260,1",

"iqn.2003-01.org.linux-iscsi.localhost.x8664:wang"

]

]

##### 对指定target进行登陆

<http://192.168.50.23:8000/api/iscsitargets/1/iscsitgt_login>

POST

{}

"ISCSI\_SUCCESS - command executed successfully."

##### 对指定target logout

<http://192.168.50.23:8000/api/iscsitargets/1/iscsitgt_logout>

POST

{}

"ISCSI\_SUCCESS - command executed successfully."

##### 列出target里的iscsi相关变量值

<http://192.168.50.23:8000/api/iscsitargets/1/iscsitgt_detail>

GET

[

"# BEGIN RECORD 6.2.0.873-33.2",

"node.name = iqn.2003-01.org.linux-iscsi.localhost.x8664:wang",

"node.tpgt = 1",

"node.startup = automatic",

"node.leading\_login = No",

"iface.hwaddress = <empty>",

"iface.ipaddress = <empty>",

"iface.iscsi\_ifacename = default",

"iface.net\_ifacename = <empty>",

"iface.transport\_name = tcp",

"iface.initiatorname = <empty>",

"iface.state = <empty>",

"iface.vlan\_id = 0",

"iface.vlan\_priority = 0",

"iface.vlan\_state = <empty>",

"iface.iface\_num = 0",

"iface.mtu = 0",

"iface.port = 0",

"iface.bootproto = <empty>",

"iface.subnet\_mask = <empty>",

"iface.gateway = <empty>",

"iface.dhcp\_alt\_client\_id\_state = <empty>",

"iface.dhcp\_alt\_client\_id = <empty>",

"iface.dhcp\_dns = <empty>",

"iface.dhcp\_learn\_iqn = <empty>",

"iface.dhcp\_req\_vendor\_id\_state = <empty>",

"iface.dhcp\_vendor\_id\_state = <empty>",

"iface.dhcp\_vendor\_id = <empty>",

"iface.dhcp\_slp\_da = <empty>",

"iface.fragmentation = <empty>",

"iface.gratuitous\_arp = <empty>",

"iface.incoming\_forwarding = <empty>",

"iface.tos\_state = <empty>",

"iface.tos = 0",

"iface.ttl = 0",

"iface.delayed\_ack = <empty>",

"iface.tcp\_nagle = <empty>",

"iface.tcp\_wsf\_state = <empty>",

"iface.tcp\_wsf = 0",

"iface.tcp\_timer\_scale = 0",

"iface.tcp\_timestamp = <empty>",

"iface.redirect = <empty>",

"iface.def\_task\_mgmt\_timeout = 0",

"iface.header\_digest = <empty>",

"iface.data\_digest = <empty>",

"iface.immediate\_data = <empty>",

"iface.initial\_r2t = <empty>",

"iface.data\_seq\_inorder = <empty>",

"iface.data\_pdu\_inorder = <empty>",

"iface.erl = 0",

"iface.max\_receive\_data\_len = 0",

"iface.first\_burst\_len = 0",

"iface.max\_outstanding\_r2t = 0",

"iface.max\_burst\_len = 0",

"iface.chap\_auth = <empty>",

"iface.bidi\_chap = <empty>",

"iface.strict\_login\_compliance = <empty>",

"iface.discovery\_auth = <empty>",

"iface.discovery\_logout = <empty>",

"node.discovery\_address = 192.168.50.172",

"node.discovery\_port = 3260",

"node.discovery\_type = send\_targets",

"node.session.initial\_cmdsn = 0",

"node.session.initial\_login\_retry\_max = 8",

"node.session.xmit\_thread\_priority = -20",

"node.session.cmds\_max = 128",

"node.session.queue\_depth = 32",

"node.session.nr\_sessions = 1",

"node.session.auth.authmethod = None",

"node.session.auth.username = <empty>",

"node.session.auth.password = <empty>",

"node.session.auth.username\_in = <empty>",

"node.session.auth.password\_in = <empty>",

"node.session.timeo.replacement\_timeout = 120",

"node.session.err\_timeo.abort\_timeout = 15",

"node.session.err\_timeo.lu\_reset\_timeout = 30",

"node.session.err\_timeo.tgt\_reset\_timeout = 30",

"node.session.err\_timeo.host\_reset\_timeout = 60",

"node.session.iscsi.FastAbort = Yes",

"node.session.iscsi.InitialR2T = No",

"node.session.iscsi.ImmediateData = Yes",

"node.session.iscsi.FirstBurstLength = 262144",

"node.session.iscsi.MaxBurstLength = 16776192",

"node.session.iscsi.DefaultTime2Retain = 0",

"node.session.iscsi.DefaultTime2Wait = 2",

"node.session.iscsi.MaxConnections = 1",

"node.session.iscsi.MaxOutstandingR2T = 1",

"node.session.iscsi.ERL = 0",

"node.conn[0].address = 192.168.50.172",

"node.conn[0].port = 3260",

"node.conn[0].startup = manual",

"node.conn[0].tcp.window\_size = 524288",

"node.conn[0].tcp.type\_of\_service = 0",

"node.conn[0].timeo.logout\_timeout = 15",

"node.conn[0].timeo.login\_timeout = 15",

"node.conn[0].timeo.auth\_timeout = 45",

"node.conn[0].timeo.noop\_out\_interval = 5",

"node.conn[0].timeo.noop\_out\_timeout = 5",

"node.conn[0].iscsi.MaxXmitDataSegmentLength = 0",

"node.conn[0].iscsi.MaxRecvDataSegmentLength = 262144",

"node.conn[0].iscsi.HeaderDigest = None",

"node.conn[0].iscsi.IFMarker = No",

"node.conn[0].iscsi.OFMarker = No",

"# END RECORD"

]

##### 修改制定target的iscsi 变量值

<http://192.168.50.23:8000/api/iscsitargets/1/iscsitgt_setvalue>

POST

{

“name”:” node.session.auth.authmethod”,

“value”:”CHAP”

}

"ISCSI\_SUCCESS - command executed successfully."

#### SCSI总线扫描

<http://192.168.40.23:8000/api/hosts/scsirescan>

#### 列表

<http://192.168.40.23:8000/api/disks>

GET

{

"count": 4,

"next": null,

"previous": null,

"results": [

{

"name": "Virtual\_disk pci-0000\_00\_10\_0-scsi-0\_0\_0\_0",

"url": "<http://192.168.40.23:8000/api/disks/1>",

"id": 1,

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"size": {

"size\_text": "30.00GB",

"size": 30720

},

"volume": "<http://192.168.40.23:8000/api/volumes/1>",

"native": {

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"rpm": 0,

"model": "Virtual\_disk",

"type": "SATA",

"serial": "pci-0000\_00\_10\_0-scsi-0\_0\_0\_0"

}

},

{

"name": "Virtual\_disk pci-0000\_00\_10\_0-scsi-0\_0\_1\_0",

"url": "<http://192.168.40.23:8000/api/disks/29>",

"id": 29,

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"size": {

"size\_text": "8.00GB",

"size": 8192

},

"volume": "<http://192.168.40.23:8000/api/volumes/29>",

"native": {

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"rpm": 0,

"model": "Virtual\_disk",

"type": "SATA",

"serial": "pci-0000\_00\_10\_0-scsi-0\_0\_1\_0"

}

},

{

"name": "Virtual\_disk pci-0000\_00\_10\_0-scsi-0\_0\_2\_0",

"url": "<http://192.168.40.23:8000/api/disks/30>",

"id": 30,

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"size": {

"size\_text": "8.00GB",

"size": 8192

},

"volume": "<http://192.168.40.23:8000/api/volumes/30>",

"native": {

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"rpm": 0,

"model": "Virtual\_disk",

"type": "SATA",

"serial": "pci-0000\_00\_10\_0-scsi-0\_0\_2\_0"

}

},

{

"name": "Virtual\_disk pci-0000\_00\_10\_0-scsi-0\_0\_3\_0",

"url": "<http://192.168.40.23:8000/api/disks/31>",

"id": 31,

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"size": {

"size\_text": "8.00GB",

"size": 8192

},

"volume": "<http://192.168.40.23:8000/api/volumes/31>",

"native": {

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"rpm": 0,

"model": "Virtual\_disk",

"type": "SATA",

"serial": "pci-0000\_00\_10\_0-scsi-0\_0\_3\_0"

}

}

]

}

# 存储池管理

## 列表

<http://192.168.40.23:8000/api/pools>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/pools/45>",

"id": 45,

"name": "test",

"uuid": "c35k9k3-2312-233d-83432",

"createdate": null,

"source\_pool": null,

"volumes": "<http://192.168.40.23:8000/api/pools/45/volumes>",

"filesystems": "<http://192.168.40.23:8000/api/pools/45/filesystems>",

"usage": {

"max\_new\_fsv": 17500,

"max\_new\_bv\_text": "15.10GB",

"usable": 15775.4,

"used": 313.0,

"usable\_text": "15.41GB",

"used\_text": "313.00MB",

"max\_new\_bv": 15462.4,

"used\_pcnt": 1.9841018294306325,

"free": 15462.4,

"size\_text": "15.41GB",

"steal\_text": "0.00B",

"free\_text": "15.10GB",

"max\_new\_fsv\_text": "17.09GB",

"steal": 0,

"size": 15775.4

},

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"native": {

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"type": {

"url": "<http://192.168.40.23:8000/api/contenttypes/24>",

"app\_label": "zfs",

"model": "zpool"

}

}

}

]

}

## 创建存储池

<http://192.168.40.23:8000/api/pools>

POST

{

"disks" : [2,3,4],

"name" : "test",

"options":{

"type" : "zfs",

“raidtype”:”raidz”,

"name" : "test"

}

}

Raidtype: raidz,mirror,raid0

## 在存储池中创建块设备卷

http://192.168.40.23:8000/api/pools/45/createvolume

POST

{

"name":"testvol",

"megs":"100",

"filesystem": ""

}

## 在存储池中创建文件卷

http://192.168.40.23:8000/api/pools/45/createvolume

POST

{

"name":"testfs",

"megs":"200",

"filesystem": "zfs"

}

成功返回：

{

"url": "<http://192.168.40.23:8000/api/volumes/75>",

"id": 75,

"name": "testfs",

"uuid": "005a0524-5882-423d-9800-c72630f3b46c",

"createdate": "2016-06-16T09:58:47.961790Z",

"source\_pool": "<http://192.168.40.23:8000/api/pools/45>",

"storage": "<http://192.168.40.23:8000/api/volumes/75/storage>",

"logdetail": "<http://192.168.40.23:8000/api/volumes/75/logdetail>",

"snapshots": "<http://192.168.40.23:8000/api/volumes/75/snapshots>",

"snapshot": null,

"usage": {

"usable": 200.0,

"used": 0.125,

"usable\_text": "200.00MB",

"used\_text": "0.12B",

"used\_pcnt": 0.0625,

"free": 199.875,

"size\_text": "200.00MB",

"steal\_text": "0.00B",

"free\_text": "199.87MB",

"steal": 0.0,

"size": 200.0

},

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"is\_protected": false,

"native": {

"is\_loged": false,

"is\_logdev": false,

"is\_filesystemvolume": true,

"fscritical": 85,

"is\_volumepool": false,

"is\_blockvolume": false,

"host": "<http://192.168.40.23:8000/api/hosts/1>",

"is\_snapshot": false,

"is\_logfor": false,

"owner": "<http://192.168.40.23:8000/api/users/1>",

"path": "/media/test/testfs",

"type": {

"url": "<http://192.168.40.23:8000/api/contenttypes/26>",

"app\_label": "zfs",

"model": "zfs"

},

"fswarning": 75

},

"upper": null

}

## 往池中增加硬盘设备

<http://192.168.40.23:8000/api/pools/45/add>

POST

{

"disks" : [78],

"options":{

"devtype" : "cache"

}

}

devtype:cache, log,raidz,mirror

将块设备78作为cache设备加入存储池

## 从池中删除块设备

<http://192.168.40.23:8000/api/pools/45/remove>

POST

{

"disks" : [78],

"options":{

}

}

# 卷管理

## 列表

<http://192.168.40.23:8000/api/volumes>

GET

#### 查询参数，除了name,uuid,create查询外，还支持下列参数

upper\_\_isnull=true

<http://192.168.40.23:8000/api/volumes?upper__isnull=true>

将列出所有没有父节点的卷设备

<http://192.168.40.23:8000/api/volumes?upper__isnull=false>

列出所有有父节点的卷设备

snapshot\_\_isnull=true

列出所有非快照设备

snapshot\_\_isnull=false

列出所有快照设备

clonefrom\_\_isnull=false

列出所有非clone设备

clonefrom\_\_isnull=false

列出所有clone设备

参数可以组合

<http://192.168.40.23:8000/api/volumes?clonefrom__isnull=true&snapshot__isnull=true&upper__isnull=true>

列出所有非子节点设备并且是非clone的设备并过滤掉快照

## 列出卷的快照

<http://192.168.40.23:8000/api/volumes/52/snapshots>

GET

## 创建卷的快照

<http://192.168.40.23:8000/api/volumes/52/snapshots>

POST

{

"name":"testsnap"

}

## 删除卷的快照

http://192.168.40.23:8000/api/snapshots/77

DELETE

## 快照卷的clone

<http://192.168.40.23:8000/api/snapshots/77>/clone

POST

{

“name”:”testclone”

}

# 客户端管理

## 列表列出所有已注册客户端

<http://192.168.40.23:8000/api/agents>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"id": 6,

"name": "192.168.40.33",

"url": "<http://192.168.40.23:8000/api/agents/6>",

"ostype": "",

"agentgroup\_set": [],

"connect\_set": [

"<http://192.168.40.23:8000/api/connects/2>"

],

"initiator\_set": [],

"oldgrains": "{'biosversion': '6.00', 'kernel': 'Linux', 'domain': '', 'uid': 0, 'zmqversion': '4.1.5', 'kernelrelease': '3.10.0-327.10.1.el7.x86\_64', 'selinux': {'enforced': 'Disabled', 'enabled': False}, 'serialnumber': 'VMware-42 1c 04 bd 33 ff dc 34-60 32 8c 6a 7b 50 19 e3', 'pid': 32761, 'ip\_interfaces': {'lo': ['127.0.0.1', '::1'], 'ens160': ['192.168.40.33', 'fe80::250:56ff:fe9c:103b'], 'virbr0-nic': [], 'virbr0': ['192.168.122.1']}, 'groupname': 'root', 'shell': '/bin/bash', 'mem\_total': 3791, 'saltversioninfo': [2016, 3, 2, 0], 'host': '', 'SSDs': ['zd0', 'dm-2', 'zd16', 'zd32', 'zd48', 'zd64', 'zd80', 'zd96'], 'mdadm': [], 'id': '192.168.40.33', 'osrelease': '7.2.1511', 'ps': 'ps -efH', 'systemd': {'version': '219', 'features': '+PAM +AUDIT +SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT +GNUTLS +ACL +XZ -LZ4 -SECCOMP +BLKID +ELFUTILS +KMOD +IDN'}, 'uuid': '421c04bd-33ff-dc34-6032-8c6a7b5019e3', 'ip6\_interfaces': {'lo': ['::1'], 'ens160': ['fe80::250:56ff:fe9c:103b'], 'virbr0-nic': [], 'virbr0': []}, 'num\_cpus': 4, 'hwaddr\_interfaces': {'lo': '00:00:00:00:00:00', 'ens160': '00:50:56:9c:10:3b', 'virbr0-nic': '52:54:00:cf:11:30', 'virbr0': '52:54:00:cf:11:30'}, 'init': 'systemd', 'ip4\_interfaces': {'lo': ['127.0.0.1'], 'ens160': ['192.168.40.33'], 'virbr0-nic': [], 'virbr0': ['192.168.122.1']}, 'osfullname': 'CentOS Linux', 'gid': 0, 'master': '192.168.40.23', 'ipv4': ['127.0.0.1', '192.168.40.33', '192.168.122.1'], 'dns': {'nameservers': ['192.168.200.1'], 'domain': '', 'ip4\_nameservers': ['192.168.200.1'], 'search': [], 'ip6\_nameservers': []}, 'ipv6': ['::1', 'fe80::250:56ff:fe9c:103b'], 'cpu\_flags': ['fpu', 'vme', 'de', 'pse', 'tsc', 'msr', 'pae', 'mce', 'cx8', 'apic', 'sep', 'mtrr', 'pge', 'mca', 'cmov', 'pat', 'pse36', 'clflush', 'dts', 'mmx', 'fxsr', 'sse', 'sse2', 'ss', 'ht', 'syscall', 'nx', 'rdtscp', 'lm', 'constant\_tsc', 'arch\_perfmon', 'pebs', 'bts', 'nopl', 'xtopology', 'tsc\_reliable', 'nonstop\_tsc', 'aperfmperf', 'eagerfpu', 'pni', 'pclmulqdq', 'ssse3', 'fma', 'cx16', 'sse4\_1', 'sse4\_2', 'movbe', 'popcnt', 'aes', 'xsave', 'avx', 'hypervisor', 'lahf\_lm', 'arat', 'pln', 'pts', 'dtherm', 'xsaveopt'], 'localhost': 'localhost', 'lsb\_distrib\_id': 'CentOS Linux', 'username': 'root', 'fqdn\_ip4': [], 'fqdn\_ip6': [], 'nodename': 'localhost', 'saltversion': '2016.3.2', 'pythonpath': ['/root/opencdp/bin', '/root/opencdp/lib64/python27.zip', '/root/opencdp/lib64/python2.7', '/root/opencdp/lib64/python2.7/plat-linux2', '/root/opencdp/lib64/python2.7/lib-tk', '/root/opencdp/lib64/python2.7/lib-old', '/root/opencdp/lib64/python2.7/lib-dynload', '/usr/lib64/python2.7', '/usr/lib/python2.7', '/root/opencdp/lib/python2.7/site-packages', '/usr/lib64/python2.7/site-packages', '/usr/lib64/python2.7/site-packages/gtk-2.0', '/usr/lib/python2.7/site-packages'], 'server\_id': 1045129586, 'saltpath': '/root/opencdp/lib/python2.7/site-packages/salt', 'pythonexecutable': '/root/opencdp/bin/python', 'osmajorrelease': '7', 'os\_family': 'RedHat', 'oscodename': 'CentOS Linux 7 (Core)', 'osfinger': 'CentOS Linux-7', 'pythonversion': [2, 7, 5, 'final', 0], 'manufacturer': 'VMware, Inc.', 'num\_gpus': 1, 'virtual': 'VMWare', 'disks': ['fd0', 'sda', 'sdb', 'sdc', 'sdd', 'sde', 'sdf', 'sdg', 'dm-0', 'dm-1'], 'cpu\_model': 'Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz', 'fqdn': '', 'biosreleasedate': '07/30/2013', 'productname': 'VMware Virtual Platform', 'osarch': 'x86\_64', 'cpuarch': 'x86\_64', 'lsb\_distrib\_codename': 'CentOS Linux 7 (Core)', 'osrelease\_info': [7, 2, 1511], 'locale\_info': {'detectedencoding': 'UTF-8', 'defaultlanguage': 'zh\_CN', 'defaultencoding': 'UTF-8'}, 'gpus': [{'model': 'SVGA II Adapter', 'vendor': 'unknown'}], 'path': '/root/opencdp/bin:/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/bin:/sbin', 'machine\_id': '07ffa01f9e964427a7e9ea2a8717e91f', 'os': 'CentOS'}"

}

]

}

## 列出所有可用客户端包括未注册的

<http://192.168.40.23:8000/api/connects/listall>

GET

{

"deny": [],

"unaccept": [],

"accept": [

"192.168.40.33",

"[WIN7-029.zktx.com](http://win7-029.zktx.com/)"

],

"reject": []

}

## 注册一个客户端

<http://192.168.40.23:8000/api/connects/register>

POST

{

"saltid":"192.168.40.33"

}

saltid填上节列出的值，如unaccept的一个值或accept里的一个值

## 判断连接是否online

<http://192.168.40.23:8000/api/connects/2/online>

GET

{

"online": true

}

## 通过一个连接获取相关agent的操作系统相关信息

<http://192.168.40.23:8000/api/connects/2/grains>

<http://192.168.40.23:8000/api/agents/6/grains>

GET

{

"biosversion": "6.00",

"kernel": "Linux",

"domain": "",

"uid": 0,

"zmqversion": "4.1.5",

"kernelrelease": "3.10.0-327.10.1.el7.x86\_64",

"selinux": {

"enforced": "Disabled",

"enabled": false

},

"serialnumber": "VMware-42 1c 04 bd 33 ff dc 34-60 32 8c 6a 7b 50 19 e3",

"pid": 32761,

"ip\_interfaces": {

"lo": [

"127.0.0.1",

"::1"

],

"ens160": [

"192.168.40.33",

"fe80::250:56ff:fe9c:103b"

],

"virbr0-nic": [],

"virbr0": [

"192.168.122.1"

]

},

"groupname": "root",

"shell": "/bin/bash",

"mem\_total": 3791,

"saltversioninfo": [

2016,

3,

2,

0

],

"host": "",

"SSDs": [

"zd0",

"dm-2",

"zd16",

"zd32",

"zd48",

"zd64",

"zd80",

"zd96"

],

"mdadm": [],

"id": "192.168.40.33",

"osrelease": "7.2.1511",

"ps": "ps -efH",

"systemd": {

"version": "219",

"features": "+PAM +AUDIT +SELINUX +IMA -APPARMOR +SMACK +SYSVINIT +UTMP +LIBCRYPTSETUP +GCRYPT +GNUTLS +ACL +XZ -LZ4 -SECCOMP +BLKID +ELFUTILS +KMOD +IDN"

},

"uuid": "421c04bd-33ff-dc34-6032-8c6a7b5019e3",

"ip6\_interfaces": {

"lo": [

"::1"

],

"ens160": [

"fe80::250:56ff:fe9c:103b"

],

"virbr0-nic": [],

"virbr0": []

},

"num\_cpus": 4,

"hwaddr\_interfaces": {

"lo": "00:00:00:00:00:00",

"ens160": "00:50:56:9c:10:3b",

"virbr0-nic": "52:54:00:cf:11:30",

"virbr0": "52:54:00:cf:11:30"

},

"init": "systemd",

"ip4\_interfaces": {

"lo": [

"127.0.0.1"

],

"ens160": [

"192.168.40.33"

],

"virbr0-nic": [],

"virbr0": [

"192.168.122.1"

]

},

"osfullname": "CentOS Linux",

"gid": 0,

"master": "192.168.40.23",

"ipv4": [

"127.0.0.1",

"192.168.40.33",

"192.168.122.1"

],

"dns": {

"nameservers": [

"192.168.200.1"

],

"domain": "",

"ip4\_nameservers": [

"192.168.200.1"

],

"search": [],

"ip6\_nameservers": []

},

"ipv6": [

"::1",

"fe80::250:56ff:fe9c:103b"

],

"cpu\_flags": [

"fpu",

"vme",

"de",

"pse",

"tsc",

"msr",

"pae",

"mce",

"cx8",

"apic",

"sep",

"mtrr",

"pge",

"mca",

"cmov",

"pat",

"pse36",

"clflush",

"dts",

"mmx",

"fxsr",

"sse",

"sse2",

"ss",

"ht",

"syscall",

"nx",

"rdtscp",

"lm",

"constant\_tsc",

"arch\_perfmon",

"pebs",

"bts",

"nopl",

"xtopology",

"tsc\_reliable",

"nonstop\_tsc",

"aperfmperf",

"eagerfpu",

"pni",

"pclmulqdq",

"ssse3",

"fma",

"cx16",

"sse4\_1",

"sse4\_2",

"movbe",

"popcnt",

"aes",

"xsave",

"avx",

"hypervisor",

"lahf\_lm",

"arat",

"pln",

"pts",

"dtherm",

"xsaveopt"

],

"localhost": "localhost",

"lsb\_distrib\_id": "CentOS Linux",

"username": "root",

"fqdn\_ip4": [],

"fqdn\_ip6": [],

"nodename": "localhost",

"saltversion": "2016.3.2",

"pythonpath": [

"/root/opencdp/bin",

"/root/opencdp/lib64/python27.zip",

"/root/opencdp/lib64/python2.7",

"/root/opencdp/lib64/python2.7/plat-linux2",

"/root/opencdp/lib64/python2.7/lib-tk",

"/root/opencdp/lib64/python2.7/lib-old",

"/root/opencdp/lib64/python2.7/lib-dynload",

"/usr/lib64/python2.7",

"/usr/lib/python2.7",

"/root/opencdp/lib/python2.7/site-packages",

"/usr/lib64/python2.7/site-packages",

"/usr/lib64/python2.7/site-packages/gtk-2.0",

"/usr/lib/python2.7/site-packages"

],

"server\_id": 1045129586,

"saltpath": "/root/opencdp/lib/python2.7/site-packages/salt",

"pythonexecutable": "/root/opencdp/bin/python",

"osmajorrelease": "7",

"os\_family": "RedHat",

"oscodename": "CentOS Linux 7 (Core)",

"osfinger": "CentOS Linux-7",

"pythonversion": [

2,

7,

5,

"final",

0

],

"manufacturer": "VMware, Inc.",

"num\_gpus": 1,

"virtual": "VMWare",

"disks": [

"fd0",

"sda",

"sdb",

"sdc",

"sdd",

"sde",

"sdf",

"sdg",

"dm-0",

"dm-1"

],

"cpu\_model": "Intel(R) Core(TM) i3-4150 CPU @ 3.50GHz",

"fqdn": "",

"biosreleasedate": "07/30/2013",

"productname": "VMware Virtual Platform",

"osarch": "x86\_64",

"cpuarch": "x86\_64",

"lsb\_distrib\_codename": "CentOS Linux 7 (Core)",

"osrelease\_info": [

7,

2,

1511

],

"locale\_info": {

"detectedencoding": "UTF-8",

"defaultlanguage": "zh\_CN",

"defaultencoding": "UTF-8"

},

"gpus": [

{

"model": "SVGA II Adapter",

"vendor": "unknown"

}

],

"path": "/root/opencdp/bin:/usr/lib64/qt-3.3/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/bin:/sbin",

"machine\_id": "07ffa01f9e964427a7e9ea2a8717e91f",

"os": "CentOS"

}

## 通过连接取到的值存入agent的oldgrains字段

<http://192.168.40.23:8000/api/connects/2/savegrains>

<http://192.168.40.23:8000/api/agents/6/savegrains>

GET

## 通过连接获取系统的磁盘情况

<http://192.168.40.23:8000/api/connects/2/diskusage>

<http://192.168.40.23:8000/api/agents/6/diskusage>

GET

{

"/media/asdfd": {

"available": "8062720",

"1K-blocks": "8062720",

"used": "0",

"capacity": "0%",

"filesystem": "asdfd"

},

"/media/test": {

"available": "23369088",

"1K-blocks": "23369088",

"used": "0",

"capacity": "0%",

"filesystem": "test"

},

"/media/test/.snapshots": {

"available": "23369088",

"1K-blocks": "23369088",

"used": "0",

"capacity": "0%",

"filesystem": "test/.snapshots"

},

"/dev": {

"available": "1925912",

"1K-blocks": "1925912",

"used": "0",

"capacity": "0%",

"filesystem": "devtmpfs"

},

"/boot": {

"available": "294288",

"1K-blocks": "508588",

"used": "214300",

"capacity": "43%",

"filesystem": "/dev/sda1"

},

"/sys/fs/cgroup": {

"available": "1941228",

"1K-blocks": "1941228",

"used": "0",

"capacity": "0%",

"filesystem": "tmpfs"

},

"/": {

"available": "10295232",

"1K-blocks": "28784916",

"used": "18489684",

"capacity": "65%",

"filesystem": "/dev/mapper/centos-root"

},

"/run": {

"available": "1923768",

"1K-blocks": "1941228",

"used": "17460",

"capacity": "1%",

"filesystem": "tmpfs"

},

"/media/test/1212": {

"available": "12160",

"1K-blocks": "12288",

"used": "128",

"capacity": "2%",

"filesystem": "test/1212"

},

"/media/test/testfs": {

"available": "102272",

"1K-blocks": "102400",

"used": "128",

"capacity": "1%",

"filesystem": "test/testfs"

},

"/media/test/zfs123": {

"available": "113536",

"1K-blocks": "113664",

"used": "128",

"capacity": "1%",

"filesystem": "test/zfs123"

},

"/run/user/0": {

"available": "388228",

"1K-blocks": "388248",

"used": "20",

"capacity": "1%",

"filesystem": "tmpfs"

},

"/media/test/tanglinhai123": {

"available": "32640",

"1K-blocks": "32768",

"used": "128",

"capacity": "1%",

"filesystem": "test/tanglinhai123"

},

"/dev/shm": {

"available": "1941116",

"1K-blocks": "1941228",

"used": "112",

"capacity": "1%",

"filesystem": "tmpfs"

}

### 客户端的initiator管理

#### 列表

<http://192.168.40.23:8000/api/initiators>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/initiators/1>",

"id": 1,

"agent": "<http://192.168.40.23:8000/api/agents/1>",

"wwn": "iqn.1991-05.com.microsoft:nodea",

"type": "iscsi"

}

]

}

#### 创建initiator

<http://192.168.40.23:8000/api/initiators>

POST

{

"wwn":"iqn.1991-05.com.microsoft:nodea"

"type":"iscsi",

"agent":1

}

type：iscsi 或者qla2xxx

### 映射

#### 列表

<http://192.168.40.23:8000/api/luns>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/luns/11>",

"id": 11,

"agent": "<http://192.168.40.23:8000/api/agents/1>",

"volume": "<http://192.168.40.23:8000/api/volumes/61>",

"lun\_id": 0,

"portals": [

"<http://192.168.40.23:8000/api/portals/1>",

"<http://192.168.40.23:8000/api/portals/3>"

]

}

]

}

#### 将块设备卷映射给某个agent

<http://192.168.40.23:8000/api/luns>

POST

api/luns post

{

"volume": 61,

"lun\_id": 0,

"agent":1,

"portals":[1,3]

}

#### 删除映射

<http://192.168.40.23:8000/api/luns/11>

DELETE

### 客户端脚本插件管理

#### 所有脚本列表：

<http://192.168.50.23:8000/api/scripts>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/scripts/1>",

"id": 1,

"name": "test",

"agent": "<http://192.168.50.23:8000/api/agents/8>", #agent url

"scriptcontent": "echo \"hello world!\"", #脚本的内容

"shelltype": "/bin/sh", #脚本类型

"when": "beforesnap", #脚本运行类型

"runas": "", #脚本运行身份

"passwd": "", #密码，windows有效

"args": "" #脚本运行参数

}

]

}

脚本类型：”/bin/sh”,”powershell”,”python”

脚本运行类型： beforesnap , aftersnap

#### 创建脚本：

<http://192.168.50.23:8000/api/scripts>

POST

{

"name": "test",

"agent": "http://192.168.50.23:8000/api/agents/8",

"scriptcontent": "echo \"hello world!\"",

"shelltype": "/bin/sh",

"when": "beforesnap",

"runas": "",

"passwd": "",

"args": ""

}

#### 更新脚本：

<http://192.168.50.23:8000/api/scripts/1>

PUT

{

"url": "http://192.168.50.23:8000/api/scripts/1",

"id": 1,

"name": "test",

"agent": "http://192.168.50.23:8000/api/agents/8",

"scriptcontent": "echo \"hello world!\"",

"shelltype": "/bin/sh",

"when": "beforesnap",

"runas": "",

"passwd": "",

"args": ""

}

#### 删除脚本

<http://192.168.50.23:8000/api/scripts/1>

DELETE

#### 运行脚本

脚本将在agent端执行

<http://192.168.50.23:8000/api/scripts/1/run>

GET

{

"pid": 21128, 脚本运行的pid

"retcode": 0, #脚本返回值

"stderr": "", #错误输出

"stdout": "hello wii\r\nopencdp" #脚步标准输出

}

# 日志设备

## 列表：

<http://192.168.40.23:8000/api/logdev>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/logdev/41>",

"id": 41,

"sourcevolume": "<http://192.168.40.23:8000/api/volumes/51>",

"logvolume": "<http://192.168.40.23:8000/api/volumes/52>",

"volume": "<http://192.168.40.23:8000/api/volumes/74>",

"usage": {

"entrys": "4134322",

"logdirs": "95",

"logdriver": "log-writes",

"logmode": "immediat",

"lastdir": "98",

"logcapcity": "204800",

"firstdir": "4"，

“lasttime”: “0”

}

}

]

}

## 给块设备包开启日志设备

<http://192.168.40.23:8000/api/logdev>

POST

{

"name":"logdev",

"sourcevolume": 51,

"logvolume": 52

}

## 察看日志

<http://192.168.40.23:8000/api/volumes/52/logdetail>

GET

## 给日志设备创建一个标志

<http://192.168.40.23:8000/api/logdev/41/mark>

POST

{

"markmsg":"mark\_a\_MARK"

}

## 在一个块设备上回放指定日志记录

<http://192.168.40.23:8000/api/volumes/61/replayto>

POST

{

"logdev":52,

"startdir":-1, #起始快照的dirno

"limit":1

“endtime”: 1498985866 #回放的时间戳

}

返回任务id

**HTTP 200 OK**

**Allow:** POST, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

{

"task\_id": "dd11504b-75ee-4495-a669-89c3839475f1"

}

## 通过此id可查询此任务的详细执行情况

# RAID卷管理

可以将两个块设备卷绑定为RAID卷，例如一个物理卷和一个zvol卷绑定为一个RAID1卷

## RAID1的列表

<http://192.168.50.23:8000/api/raid1>

GET

{

"url": "<http://192.168.50.23:8000/api/raid1/82>",

"id": 82,

"sourcevolume": "<http://192.168.50.23:8000/api/volumes/11>",

"dstvolume": "<http://192.168.50.23:8000/api/volumes/12>",

"volume": "<http://192.168.50.23:8000/api/volumes/130>",

"usage": [

"0",

"2048",

"mirror",

"2",

"230:6496",

"230:8224",

"2/2", #同步的进度分子表示已同步数，分母表示总共需要同步数量

"1",

"AA",

"1",

"core"

]

}

硬盘状态

\* A => Alive - No failures

\* D => Dead - A write failure occurred leaving mirror out-of-sync

\* S => Sync - A sychronization failure occurred, mirror out-of-sync

\* R => Read - A read failure occurred, mirror data unaffected

## MirrorVG创建卷组镜像

<http://192.168.50.23:8000/api/agentmirror>

GET

**HTTP 200 OK**

**Allow:** GET, POST, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.62:8000/api/agentmirror/1>",

"id": 1,

"agentname": "192.168.70.252",

"vgname": "rootvg",

"vgdetails": "hdisk0 active 273 8 00..00..00..00..08\n"

},

{

"url": "<http://192.168.50.62:8000/api/agentmirror/13>",

"id": 13,

"agentname": "192.168.70.252",

"vgname": "datavg",

"vgdetails": "hdisk9 active 4341 4341 869..868..868..868..868\n"

}

]

}

POST

{

"agentname":"192.168.70.252", #客户端的名称

"vgname":"datavg", #卷组名

"sourcedisk": "hdisk9", #源卷

"destdisk":"hdisk10", #目标卷

"size":8 #物理卷的大小

}

[**http://192.168.50.62:8000/api/agentmirror/13/refresh**](http://192.168.50.62:8000/api/agentmirror/13/refresh)

POST

{

"agentname":"192.168.70.252", #客户端的名称

"vgname":"datavg" #卷组名

}

<http://192.168.50.62:8000/api/agentmirror/13/remove>

POST

{

"agentname":"192.168.70.252", #客户端的名称

"vgname":"datavg" #卷组名

"volumename":"hdisk10", #卷名

"islast":True #是不是最后一个

}

## MirrorDisk创建卷组镜像

<http://192.168.50.62:8000/api/agentmirrorhp>

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.62:8000/api/agentmirrorhp/2>",

"id": 2,

"agentname": "192.168.70.58",

"vgname": "/dev/vg00",

"vgdetails": "[\"/dev/disk/disk2\_p2\", \"/dev/disk/disk5\"]"

}

]

}

<http://192.168.50.62:8000/api/agentmirrorhp/2/unmirror>

POST

{

"agentname": "192.168.70.58",

"vgname": "/dev/vg00",

"diskname": "/dev/disk/disk5"

}

<http://192.168.50.62:8000/api/agentmirrorhp/2/mirror>

POST

{

"agentname": "192.168.70.58",

"vgname": "/dev/vg00",

"diskname": "/dev/disk/disk5"

}

<http://192.168.50.62:8000/api/agentmirror/13/remove>

POST

{

"agentname": "192.168.70.58",

"vgname": "/dev/vg00",

"diskname": "/dev/disk/disk5"

"islast":True #是不是最后一个

}

## 删除MirrorVG卷

卷组最后一个物理磁盘remove之后,自动删除

## RAID1创建

<http://192.168.50.23:8000/api/raid1>

POST

{

"name":"raidtest", #卷名

"sourcevolume": 11, #源卷

"dstvolume":12, #目标卷

"sync":true #创建的时候是否开始同步

}

<http://192.168.50.62:8000/api/raid1/30/loadresume>

POST

{

"sourcevolume": 11, #源卷

"dstvolume":12, #目标卷

"sync":true #创建的时候是否开始同步

}

{

"url": "<http://192.168.50.23:8000/api/volumes/130>",

"id": 130,

"name": "raidtest",

"uuid": "d1ab031f-fed7-4f47-acfd-79a6597b365d",

"createdate": "2017-06-23T11:03:20Z",

"source\_pool": null,

"storage": "<http://192.168.50.23:8000/api/volumes/130/storage>",

"logdetail": "<http://192.168.50.23:8000/api/volumes/130/logdetail>",

"snapshots": "<http://192.168.50.23:8000/api/volumes/130/snapshots>",

"snapshot": null,

"usage": {

"size\_text": "1.00MB",

"size": 1

},

"status": {

"status": "good",

"text": "Everything seems to be in order",

"flags": {

"online": "The volume is accessible."

}

},

"is\_protected": false,

"native": {

"is\_loged": false,

"is\_logdev": false,

"raid": {

"url": "<http://192.168.50.23:8000/api/raid1/82>",

"id": 82,

"sourcevolume": "<http://192.168.50.23:8000/api/volumes/11>",

"dstvolume": "<http://192.168.50.23:8000/api/volumes/12>",

"volume": "<http://192.168.50.23:8000/api/volumes/130>",

"usage": [

"0",

"2048",

"mirror",

"2",

"230:6496",

"230:8224",

"2/2",

"1",

"AA",

"1",

"core"

]

},

"is\_filesystemvolume": false,

"is\_raid1": true,

"fscritical": null,

"is\_volumepool": false,

"is\_blockvolume": true,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"is\_snapshot": false,

"is\_logfor": false,

"owner": null,

"path": "/dev/mapper/raidtest",

"type": {

"url": "<http://192.168.50.23:8000/api/contenttypes/77>",

"app\_label": "raid",

"model": "raid1"

},

"fswarning": null

},

"upper": null,

"clonefrom": null,

"clone\_set": []

}

## 删除RAID卷

与普通卷删除相同

# 镜像卷的管理

可以将agent端的一个块设备与cdp的一个块设备使用drbd软件创建镜像关系

## 镜像的列表

<http://192.168.40.23:8000/api/mirrors>

GET

镜像列表时可用的查询参数

agent=agentid

设置此参数将只列出指定的agent所属mirror配置

http://192.168.40.23:8000/api/mirrors?agent=6

## 创建镜像设置

<http://192.168.40.23:8000/api/mirrors>

POST

{

"protocol":"C",

"sync\_rate":"5M",

"source":{

"agentvolumename":"6\_testmirror", #镜像的名字

"agentDevpath":"/dev/zvol/test/testmirror", #agent端的块设备

"agentDrbdminor":1, #agent端的drbd设备号，

"agentaddress":"192.168.40.33", #agent端用来建立mirror关系的ip

"metadisk":"/dev/zvol/test/testmirrormeta", #drbd的用来存放meta的设备路径，如果为””值，将建立metadisk 为internal类型的drbd设备

"agentid":6 #agent的id

},

"dest":{

"drbdenable":true, #本地drbd开启，默认true

"volume\_id":190, #本地端的块设备

“metadisk”: 0, #本地端meta设备，如果为不存在的设备将建立internal的metadisk

"localaddress":3 #本地的ip编号

}

}

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"id": 91,

"url": "<http://192.168.40.23:8000/api/mirrors/91>",

"protocol": "C",

"syncer\_rate": "5M",

"port": 7710,

"name": "6\_testmirror",

"volume": "<http://192.168.40.23:8000/api/volumes/204>",

"getstatus": [

"unknown"

],

"getagentendpoint": [

{

"agentvolumename": "6\_testmirror",

"agentDevpath": "/dev/zvol/test/testmirror",

"agentDrbdminor": 1,

"agentaddress": "192.168.40.33",

"metadisk": "/dev/zvol/test/testmirrormeta",

"agent": "<http://192.168.40.23:8000/api/agents/6>",

"port": 7701

}

],

"getlocalendpoint": [

{

"volume": "<http://192.168.40.23:8000/api/volumes/203>",

"metadisk": null,

"ipaddress": "<http://192.168.40.23:8000/api/ipaddresses/3>",

"is\_primary": null,

"drbdstatus": [

"unknown",

"unknown",

"unknown"

]

}

]

}

]

}

## 创建agent -> agent的mirror设置

<http://192.168.40.23:8000/api/mirrors>

POST

{

"protocol":"C",

"sync\_rate":"5M",

"source":{

"agentvolumename":"6\_testmirror",

"agentDevpath":"/dev/zvol/test/testmirror",

"agentDrbdminor":1,

"agentaddress":"192.168.50.23",

"metadisk":"/dev/zvol/test/testmirrormeta",

"agentid":6

},

"dest":{

"volume\_id":190, 本地的volume

"localaddress":3,

"drbdenable":false, #本地drbd关闭，也就是agent -> agent开启

"agentvolumename":"7\_testmirror",

"agentDevpath":"/dev/zvol/test/testmirror",

"agentDrbdminor":1,

"agentaddress":"192.168.50.174",

"metadisk":"/dev/zvol/test/testmirrormeta",

"agentid":7

}

}

## 修改agentendpoint端信息

<http://192.168.50.23:8000/api/agentendpoints/2>

PUT

{

"id": 2,

"url": "<http://192.168.50.23:8000/api/agentendpoints/2>",

"agentvolumename": "drbd",

"agentDevpath": "/dev/zd128",

"agentDrbdminor": 12,

"agentaddress": "192.168.40.74",

"metadisk": "",

"agent": "<http://192.168.50.23:8000/api/agents/5>",

"port": 7712,

"primarybf\_cmd": "/tmp/a.sh", #执行primary前执行脚本

"primaryaf\_cmd": "/tmp/a.sh", #执行primary后执行脚本

"secondarybf\_cmd": null, #执行secondary前执行脚本

"secondaryaf\_cmd": null, #执行secondary后执行脚本

"upbf\_cmd": null, #执行up前执行脚本

"upaf\_cmd": null, #执行up后执行脚本

"downbf\_cmd": null, #执行down前执行脚本

"downaf\_cmd": null #执行down后执行脚本

}

## 将mirror设置里的配置安装drbd设备，执行后将分别在agent端和本地端执行创建drbd设备

<http://192.168.40.23:8000/api/mirrors>

POST

{

"connection\_id":91

}

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"id": 91,

"url": "<http://192.168.40.23:8000/api/mirrors/91>",

"protocol": "C",

"syncer\_rate": "5M",

"port": 7710,

"name": "6\_testmirror",

"volume": "<http://192.168.40.23:8000/api/volumes/204>",

"getstatus": [

"online"

],

"getagentendpoint": [

{

"agentvolumename": "6\_testmirror",

"agentDevpath": "/dev/zvol/test/testmirror",

"agentDrbdminor": 1,

"agentaddress": "192.168.40.33",

"metadisk": "/dev/zvol/test/testmirrormeta",

"agent": "<http://192.168.40.23:8000/api/agents/6>",

"port": 7701

}

],

"getlocalendpoint": [

{

"volume": "<http://192.168.40.23:8000/api/volumes/203>",

"metadisk": null,

"ipaddress": "<http://192.168.40.23:8000/api/ipaddresses/3>",

"is\_primary": false,

"drbdstatus": [

"degraded",

"degraded",

"secondary"

]

}

]

}

]

}

## 开启agent-〉local初始化进程

<http://192.168.40.23:8000/api/mirrors/91/primaryagent>

GET

{

"id": 91,

"url": "<http://192.168.40.23:8000/api/mirrors/91>",

"protocol": "C",

"syncer\_rate": "5M",

"port": 7710,

"name": "6\_testmirror",

"volume": "<http://192.168.40.23:8000/api/volumes/204>",

"getstatus": [

"online"

],

"getagentendpoint": [

{

"agentvolumename": "6\_testmirror",

"agentDevpath": "/dev/zvol/test/testmirror",

"agentDrbdminor": 1,

"agentaddress": "192.168.40.33",

"metadisk": "/dev/zvol/test/testmirrormeta",

"agent": "<http://192.168.40.23:8000/api/agents/6>",

"port": 7701

}

],

"getlocalendpoint": [

{

"volume": "<http://192.168.40.23:8000/api/volumes/203>",

"metadisk": null,

"ipaddress": "<http://192.168.40.23:8000/api/ipaddresses/3>",

"is\_primary": false,

"drbdstatus": [

"degraded",

"online",

"secondary"

]

}

]

}

Drbdstatus显示分别是

本地盘状态，agent盘状态，本地地位

盘状态： unknown,locked,degraded,online,offline

本地地位：unknown,primary,secondary

## 开启同步到本地方向，agent数据将开始同步到本地盘上

http://192.168.40.23:8000/api/mirrors/91/sync2local

GET

## 开启同步到远程，将本地盘数据同步到agent的盘上

http://192.168.40.23:8000/api/mirrors/91/sync2agent

GET,POST

{

‘id’: 6 #agent id

}

## 将指定agent设置为primary

<http://192.168.40.23:8000/api/mirrors/91/primaryagent>

GET,POST

{

“id”: 7 , #agent id

“bfcmd”:false, #是否执行primary前脚本

“afcmd”:false # 是否执行primary后脚本  
}

## 将agent设置为secondary

<http://192.168.40.23:8000/api/mirrors/91/secondaryagent>

GET,POST

{

‘id’: 7， #agent id

“bfcmd”:false, #是否执行secondary前脚本

“afcmd”:false # 是否执行secondary后脚本  
  
}

## 查询同步完成的进度

<http://192.168.40.23:8000/api/mirrors/91/getsyncstatus>

GET,POST

{

‘id’: 7 #agent id  
}

**HTTP 200 OK**

**Allow:** GET, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

[

"24.32"

]

如果没有正在同步，或者同步已经完成

将返回

**HTTP 200 OK**

**Allow:** GET, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

[

"unknown"

]

## 卸载drbd设备，但不删除mirror配置

http://192.168.40.23:8000/api/mirrors/91/uninstall

GET

## 只安装agent端的drbd设备

http://192.168.40.23:8000/api/mirrors/91/installagent

GET,POST

{

“id”: 7 , #agent id

“bfcmd”:false,

“afcmd”:false  
}

## 查询agent端角色

http://192.168.40.23:8000/api/mirrors/91/getagentrole

GET,POST

{

‘id’: 7 #agent id  
}

## Upagent端卷

http://192.168.40.23:8000/api/mirrors/91/upagent

GET,POST

{

‘id’: 7 #agent id

“bfcmd”:false, #是否执行up前脚本

“afcmd”:false # 是否执行up后脚本  
}

## Downagent端卷

http://192.168.40.23:8000/api/mirrors/91/downagent

GET,POST

{

‘id’: 7 #agent id

“bfcmd”:false, #是否执行down前脚本

“afcmd”:false # 是否执行down后脚本  
}

## 暂停同步

http://192.168.40.23:8000/api/mirrors/91/pausesync

GET,POST

{

‘id’: 7 #agent id  
}

## 重新开启同步

http://192.168.40.23:8000/api/mirrors/91/resumesync

GET,POST

{

‘id’: 7 #agent id  
}

## 删除mirror配置

<http://192.168.40.23:8000/api/mirrors/91>

DELETE

# 文件同步节点管理

可以设置文件同步服务，将服务器的一个文件卷设置为同步节点的目的端

## 所有同步节点列出

<http://192.168.50.23:8000/api/rsyncs>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/rsyncs/3>",

"id": 3,

"name": "testfs",

"path": "/media/test/testfs",

"comment": "test",

"volume": "<http://192.168.50.23:8000/api/volumes/24>"

}

]

}

## 增加同步节点

<http://192.168.50.23:8000/api/rsyncs>

POST

{

"name": "testfs", #节点名

"path": "/media/test/testfs", #同步的目的目录

"comment": "test", #说明

"volume": "<http://192.168.50.23:8000/api/volumes/24>" #文件卷

}

## 删除同步节点

<http://192.168.50.23:8000/api/rsyncs/3>

DELETE

## 修改

<http://192.168.50.23:8000/api/rsyncs/3>

UPDATE

# 文件克隆

将agent端的目录复制到文件同步节点上

## 列出文件克隆设置

<http://192.168.50.23:8000/api/fileclones>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/fileclones/1>",

"id": 1,

"name": "test",

"agent": "<http://192.168.50.23:8000/api/agents/5>",

"agentpath": "/tmp",

"task": "<http://192.168.50.23:8000/api/results/8>",

"localpath": "test",

"localend": "<http://192.168.50.23:8000/api/rsyncs/3>",

"localip": "<http://192.168.50.23:8000/api/ipaddresses/3>"

}

]

}

## 增加文件克隆设置

<http://192.168.50.23:8000/api/fileclones>

POST

{

"name": "test", #名称

"agent": "<http://192.168.50.23:8000/api/agents/5>", #agent

"agentpath": "/tmp", #agent端的目录

"localpath": "test", #本地端相对目录

"localend": "<http://192.168.50.23:8000/api/rsyncs/3>", #本地端的同步节点

"localip": "<http://192.168.50.23:8000/api/ipaddresses/3>" #本地端的ip

}

## 修改文件克隆设置

<http://192.168.50.23:8000/api/fileclones/1>

UPDATE

{

"name": "test", #名称

"agent": "<http://192.168.50.23:8000/api/agents/5>", #agent

"agentpath": "/tmp", #agent端的目录

"localpath": "test", #本地端相对目录

"localend": "<http://192.168.50.23:8000/api/rsyncs/3>", #本地端的同步节点

"localip": "<http://192.168.50.23:8000/api/ipaddresses/3>" #本地端的ip

}

## 删除文件克隆设置

<http://192.168.50.23:8000/api/fileclones/1>

DELETE

## 启动文件克隆从agent到local

http://192.168.50.23:8000/api/fileclones/1/sync2local

GET,POST

## 启动文件克隆从local到agent

<http://192.168.50.23:8000/api/fileclones/1/sync2agent>

GET,POST

# samba共享设置

可以将文件卷以samba协议网络共享

## 列表

<http://192.168.50.23:8000/api/sambashares>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/sambashares/5>",

"id": 5,

"name": "testfsshare", #网络共享服务名

"path": "/media/test/testfs", #本地目录

"available": true,

"browseable": true,

"guest\_ok": false,

"writeable": false,

"comment": "",#注释说明

"volume": "<http://192.168.50.23:8000/api/volumes/24>" #本地相关文件卷

}

]

}

## 增加网络共享

<http://192.168.50.23:8000/api/sambashares>

POST

{

"name": "testfsshare",

"path": "/media/test/testfs",

"available": true,

"browseable": true,

"guest\_ok": false,

"writeable": true,

"comment": "",

"volume": "<http://192.168.50.23:8000/api/volumes/24>"

}

## 删除网络共享

<http://192.168.50.23:8000/api/sambashares/5>

DELETE

# NFS网络共享

## 列表

<http://192.168.50.23:8000/api/nfs>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/nfs/10>",

"id": 10,

"name": "testv",

"path": "/media/test/testfs",

"comment": "",

"volume": "<http://192.168.50.23:8000/api/volumes/24>",

"rw": true

}

]

}

## 创建nfs共享

<http://192.168.50.23:8000/api/nfs>

POST

{

"name": "testv",

"path": "/media/test/testfs", #共享的目录

"comment": "",

"volume": "<http://192.168.50.23:8000/api/volumes/24>", #共享的文件卷

"rw": true #是否可写

}

## 修改nfs共享

<http://192.168.50.23:8000/api/nfs/10>

PUT

{

"name": "testv"  
}

## 删除nfs共享

<http://192.168.50.23:8000/api/nfs/10>

DELETE

# 物理盘复制任务

对agent物理盘通过网络复制到本地卷或相反的操作

复制任务列表

<http://192.168.40.23:8000/api/clones>

GET

{

"id": 2,

"url": "<http://192.168.40.23:8000/api/clones/2>",

"name": "clonetest",

"clonepos": 11218944,

"getagentendpoint": [

{

"agentDevpath": "/dev/zvol/test/testvol",

"agentaddress": "192.168.40.74",

"agentDevsize": 100000000,

"clonepid": null,

"agent": "<http://192.168.40.23:8000/api/agents/10>",

"port": 9000

}

],

"getlocalendpoint": [

{

"volume": "<http://192.168.40.23:8000/api/volumes/30>",

"ipaddress": "<http://192.168.40.23:8000/api/ipaddresses/3>",

"port": 9000

}

],

"task": "<http://192.168.40.23:8000/api/results/29>"

}

]

创建clone任务

<http://192.168.40.23:8000/api/clones>

POST

{

"name":"clonetest",

"source":{

"agentDevpath":"/dev/zvol/test/testvol",

"agentDevsize":100000000,

"agentaddress":"192.168.40.74",

"agentid":10,

"port":9000

},

"dest":{

"volume\_id":30,

"localaddress":3,

"port":9000

}

}

开始复制任务，agent到local

<http://192.168.40.23:8000/api/clones/2/sync2local>

POST

{

“bs”:512, #块大小，默认512

“offset”:0 #传输起始位置  
}

开始复制任务，local到agent

<http://192.168.40.23:8000/api/clones/2/sync2agent>

POST

{

“bs”:512,

“offset”:0  
}

# 长时间任务结果及调度

## 任务结果列表

<http://192.168.40.23:8000/api/results>

GET

**HTTP 200 OK**

**Allow:** GET, POST, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

{

"count": 31,

"next": "<http://192.168.40.23:8000/api/results?limit=10&offset=10>",

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/results/63>",

"id": 63,

"task\_id": "c866fecb-4c63-4af1-94ec-9ed8b88961e1",

"task": "celery.backend\_cleanup",

"status": "SUCCESS",

"native": {

"submit\_at": "2016-07-03T04:00:00.015Z",

"args": "[]",

"result": "None",

"done\_at": "2016-07-03T04:00:03.980Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/64>",

"id": 64,

"task\_id": "498d8a6c-3e14-4282-b803-5877993e8021",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T04:34:25.640Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T04:34:26.922Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/65>",

"id": 65,

"task\_id": "b8415b95-5c7e-4466-a258-697ce7da4b41",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T05:34:25.640Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T05:34:27.103Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/66>",

"id": 66,

"task\_id": "4c08f897-869b-40c1-8909-77a241c5184e",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T06:34:25.641Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T06:34:27.315Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/67>",

"id": 67,

"task\_id": "5d4a0993-d67b-4107-aa53-fcd097b98616",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T07:34:25.641Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T07:34:27.455Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/68>",

"id": 68,

"task\_id": "0c8b0054-f7c7-426c-8401-8d4669c56569",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T08:34:25.642Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T08:34:27.562Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/69>",

"id": 69,

"task\_id": "a778f6cc-8638-430a-9507-8510d60a5bd2",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T09:34:25.642Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T09:34:27.648Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/70>",

"id": 70,

"task\_id": "13c70550-71d9-4f89-9ea5-dc0b0659b853",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T10:34:25.643Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T10:34:27.814Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/71>",

"id": 71,

"task\_id": "90fc22eb-faa3-4992-9a9f-5f660e7ae186",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T11:34:25.644Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T11:34:27.950Z"

}

},

{

"url": "<http://192.168.40.23:8000/api/results/72>",

"id": 72,

"task\_id": "949fae4e-987e-4956-bb8c-137dd96f0daa",

"task": "volumes.tasks.task\_replay\_to",

"status": "FAILURE",

"native": {

"submit\_at": "2016-07-03T12:34:25.645Z",

"args": "[]",

"result": "TypeError('task\_replay\_to() takes exactly 4 arguments (0 given)',)",

"done\_at": "2016-07-03T12:34:28.059Z"

}

}

]

}

## 根据id查找任务详情

<http://192.168.40.23:8000/api/results?task_id=c866fecb-4c63-4af1-94ec-9ed8b88961e1>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/results/63>",

"id": 63,

"task\_id": "c866fecb-4c63-4af1-94ec-9ed8b88961e1",

"task": "celery.backend\_cleanup",

"status": "SUCCESS",

"native": {

"submit\_at": "2016-07-03T04:00:00.015Z",

"args": "[]",

"result": "None",

"done\_at": "2016-07-03T04:00:03.980Z"

}

}

]

}

## 创建调度任务周期调度

<http://192.168.40.23:8000/api/schedules>

POST

{

"ptask\_name":"testptask",

"task\_name":"volumes.tasks.task\_replay\_to",

"period":"hours",

"every":"1",

"args":"[\"/dev/zvol/test/testclone\", \"/dev/zvol/test/testvol2\", -1, 2]",

"kwargs":{}

}

## 创建调度任务crontab调度

<http://192.168.40.23:8000/api/schedules>

POST

{

"ptask\_name":"testptask",

"task\_name":"volumes.tasks.task\_replay\_to",

"minute":"0", #分钟 minute 0-59

"hour":"0,12", #小时，0-23

“week”:”\*”, #星期几，0-7

“day”:”\*”, #day of month 1-31

“month”:”\*”, # 月份 1-12

"args":"[\"/dev/zvol/test/testclone\", \"/dev/zvol/test/testvol2\", -1, 2]",

"kwargs":{}

}

## 修改调度任务的参数

<http://192.168.50.23:8000/api/schedules/2>

PUT

{

"name": "tgb",

"args": "[37]",

"kwargs": "{\"max\_snap\_num\": 44}",

}

只有上面三个值可以被修改

## 修改调度任务的调度时间

<http://192.168.50.23:8000/api/schedules/2/chgschedule>

POST

修改为周期调度

{

"period":"hours",

"every":"1",

}

修改成crontab类型

{

"minute":"0", #分钟 minute 0-59

"hour":"0,12", #小时，0-23

“week”:”\*”, #星期几，0-7

“day”:”\*”, #day of month 1-31

“month”:”\*”, # 月份 1-12

}

## 启动调度任务

<http://192.168.40.23:8000/api/schedules/3/start>

POST

{}

## 停止调度任务

<http://192.168.40.23:8000/api/schedules/3/stop>

POST

{}

返回

{

"url": "<http://192.168.40.23:8000/api/schedules/3>",

"id": 3,

"name": "testptask",

"task": "volumes.tasks.task\_replay\_to",

"schedule": "<IntervalSchedule: every hour>",

"args": "[\"/dev/zvol/test/testclone\", \"/dev/zvol/test/testvol2\", -1, 2]",

"kwargs": "{}",

"enabled": false

}

## 同时支持task,和status组合查找

<http://192.168.40.23:8000/api/results?task_id=&task=celery.backend_cleanup&status=SUCCESS>

GET

## 删除调度任务

<http://192.168.40.23:8000/api/schedules/3>

DELETE

## 自动创建快照任务

task\_name:

volumes.tasks.auto\_create\_snap

args: [volumeid]

kwargs:{"max\_snap\_num":200, “scripts”:[1,2], “check\_befsnap”:True}

参数说明

Volumeid：需要创建快照的块设备id

Max\_snap\_num:最大保留快照，超过这个值后将自动删除多出来的快照

scripts: 需要运行的脚本列表

check\_befsnap: 快照前是否检查快照前脚本的返回值，根据脚本返回值来决定是否创建快照，脚本返回0值代表脚本执行成功，非0表示脚本出错

## 自动文件克隆任务

task\_name:

clone.tasks.fileclone2local

args: [filecloneid]

参数说明

filecloneid：需要定时运行的fileclone设置

# 虚拟云管理

## 列出云主机配置

<http://192.168.40.23:8000/api/clouds>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.40.23:8000/api/clouds/2>",

"id": 2,

"name": "localhost",

"connection": "qemu:///system"

}

]

}

## 创建云主机连接配置

<http://192.168.40.23:8000/api/clouds>

POST

{

“name”:”localhost”,

“connection”:”qemu:///system”

}

现在只支持本地qemu虚拟机

## 创建虚拟机

<http://192.168.40.23:8000/api/clouds/2/createvpc>

POST

{

“name”:”demo”, #虚拟机名，没有则随机生成

“persist”:false #是否持久

“bootide”:false #是否第一个引导盘为IDE

“kvmtype”:”qemu”, #虚拟机格式，可以的值qemu kvm

“cpu”:2, #CPU数量，没有则默认2

“mem”:”52334”, #内存数量，单位k, 默认是1G

“volumes”:[13] #硬盘卷，

}

# 集群的管理

使用ceph作为集群管理，创建的集群卷将自动在所有集群使用节点中出现

## 集群的列表

<http://192.168.50.23:8000/api/cephclusters>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/cephclusters/61>",

"id": 61,

"name": "ceph",

"crushmap": {

"id": 6,

"crushmap": {

"tunables": {

"profile": "firefly",

"minimum\_required\_version": "firefly",

"has\_v3\_rules": 0,

"has\_v4\_buckets": 0,

"choose\_total\_tries": 50,

"require\_feature\_tunables3": 1,

"require\_feature\_tunables5": 0,

"legacy\_tunables": 0,

"chooseleaf\_descend\_once": 1,

"chooseleaf\_stable": 0,

"choose\_local\_fallback\_tries": 0,

"has\_v2\_rules": 0,

"straw\_calc\_version": 1,

"allowed\_bucket\_algs": 22,

"has\_v5\_rules": 0,

"require\_feature\_tunables2": 1,

"optimal\_tunables": 0,

"choose\_local\_tries": 0,

"chooseleaf\_vary\_r": 1,

"require\_feature\_tunables": 1

},

"types": [

{

"name": "osd",

"type\_id": 0

},

{

"name": "host",

"type\_id": 1

},

{

"name": "chassis",

"type\_id": 2

},

{

"name": "rack",

"type\_id": 3

},

{

"name": "row",

"type\_id": 4

},

{

"name": "pdu",

"type\_id": 5

},

{

"name": "pod",

"type\_id": 6

},

{

"name": "room",

"type\_id": 7

},

{

"name": "datacenter",

"type\_id": 8

},

{

"name": "region",

"type\_id": 9

},

{

"name": "root",

"type\_id": 10

}

],

"rules": [

{

"min\_size": 1,

"rule\_name": "replicated\_ruleset",

"steps": [

{

"item\_name": "default",

"item": -1,

"op": "take"

},

{

"num": 0,

"type": "host",

"op": "chooseleaf\_firstn"

},

{

"op": "emit"

}

],

"ruleset": 0,

"type": 1,

"rule\_id": 0,

"max\_size": 10

}

],

"buckets": [

{

"hash": "rjenkins1",

"name": "default",

"weight": 1022,

"type\_id": 10,

"alg": "straw",

"type\_name": "root",

"items": [

{

"hash": "rjenkins1",

"name": "centos7-2",

"weight": 511,

"type\_id": 1,

"alg": "straw",

"type\_name": "host",

"items": [],

"id": -2

},

{

"hash": "rjenkins1",

"name": "centos7-3",

"weight": 511,

"type\_id": 1,

"alg": "straw",

"type\_name": "host",

"items": [],

"id": -3

}

],

"id": -1

}

],

"devices": [

{

"id": 0,

"name": "osd.0"

},

{

"id": 1,

"name": "osd.1"

}

]

},

"epoch": 78,

"created\_at": "2017-06-16T09:10:54Z",

"edited\_at": "2017-06-16T09:10:54Z",

"cluster": 61,

"author": null

},

"mon\_set": [

"<http://192.168.50.23:8000/api/mons/1>"

],

"auth\_cluster\_required": "cephx",

"auth\_client\_required": "cephx",

"auth\_service\_required": "cephx"

}

]

## Cephpool管理

### 集群pool的列表

<http://192.168.50.23:8000/api/cephpools>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/cephpools/2>",

"id": 2,

"ceph\_id": 0,

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>",

"size": 2, #默认副本数

"min\_size": 1, #最小副本数

"ruleset": 0,

"storageobj": "<http://192.168.50.23:8000/api/pools/62>"

}

]

}

### 修改cephpool

<http://192.168.50.23:8000/api/cephpools>/2

PUT

{

“size”: 2,

“min\_size”:1

}

### 删除cephpool

<http://192.168.50.23:8000/api/cephpools>/2

DELETE

## Cephclient管理

Cephclient是集群的使用者，集群中创建的ceph块设备将自动被所有cephclient使用

### Cephclient列表

<http://192.168.50.23:8000/api/cephclients>

GET

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/cephclients/1>",

"id": 1,

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>",

"host": "<http://192.168.50.23:8000/api/hosts/3>"

},

{

"url": "<http://192.168.50.23:8000/api/cephclients/2>",

"id": 2,

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>",

"host": "<http://192.168.50.23:8000/api/hosts/1>"

}

]

}

### Cephclient增加

<http://192.168.50.23:8000/api/cephclients>

POST

{

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>", #集群url

"host": "<http://192.168.50.23:8000/api/hosts/3>" #使用的cdp主机url

}

### Cephosd 列表

<http://192.168.50.23:8000/api/osds>

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/osds/1>",

"id": 1,

"uuid": "1a1f1730-0624-41fa-bbb9-0fdaa56f51b2",

"ceph\_id": 0,

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>",

"volume": null,

"journal": null

},

{

"url": "<http://192.168.50.23:8000/api/osds/2>",

"id": 2,

"uuid": "bf7ddb78-9ea6-4a47-b6fe-704530071bc5",

"ceph\_id": 1,

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>",

"volume": null,

"journal": null

}

]

}

### Ceph osd增加

将硬盘添加到ceph集群中，成为osd节点

<http://192.168.50.23:8000/api/osds>

POST

{

"cluster": "<http://192.168.50.23:8000/api/cephclusters/61>", #cluster

"volume": "volume": "<http://192.168.50.71:8000/api/volumes/60>", #块设备，最好用物理盘

"journal": "<http://192.168.50.71:8000/api/volumes/60>" # ceph用的块设备，最好用物理盘

}

返回

{

"url": "<http://192.168.50.71:8000/api/osds/4>",

"id": 4,

"uuid": "437e8023-fbd1-4b78-acb5-39992f8b9299",

"ceph\_id": 2,

"cluster": "<http://192.168.50.71:8000/api/cephclusters/105>",

"volume": "<http://192.168.50.71:8000/api/volumes/60>",

"journal": "<http://192.168.50.71:8000/api/volumes/60>"

}

### 删除指定OSD

<http://192.168.50.71:8000/api/osds/4>

DELETE

### 其他节点更新ceph 信息

<http://192.168.50.23:8000/api/cephclusters/refresh>

GET

# 系统服务

## 获取所有服务

<http://192.168.50.23:8000/api/systemctls>

GET

**HTTP 200 OK**

**Allow:** GET, POST, HEAD, OPTIONS

**Content-Type:** application/json

**Vary:** Accept

{

"count": 7,

"next": null,

"previous": null,

"results": [

{

"id": 1,

"name": "salt-master",

"status": "Unknown",

"usage": "\* salt-master.service - The Salt Master Server\n Loaded: loaded (/usr/lib/systemd/system/salt-master.service; enabled; vendor preset: disabled)\n Active: active (running) since Fri 2017-07-21 10:19:35 UTC; 1h 27min ago\n Main PID: 1296 (/usr/bin/python)\n CGroup: /system.slice/salt-master.service\n |-1296 /usr/bin/python /usr/bin/salt-master ProcessManage\n |-2893 /usr/bin/python /usr/bin/salt-master MultiprocessingLoggingQueu\n |-2895 /usr/bin/python /usr/bin/salt-master ZeroMQPubServerChanne\n |-2898 /usr/bin/python /usr/bin/salt-master EventPublishe\n |-2899 /usr/bin/python /usr/bin/salt-master Maintenanc\n |-2900 /usr/bin/python /usr/bin/salt-master ReqServer\_ProcessManage\n |-2901 /usr/bin/python /usr/bin/salt-master MWorkerQueu\n |-2908 /usr/bin/python /usr/bin/salt-master MWorker-\n |-2909 /usr/bin/python /usr/bin/salt-master MWorker-\n |-2910 /usr/bin/python /usr/bin/salt-master MWorker-\n |-2911 /usr/bin/python /usr/bin/salt-master MWorker-\n `-2912 /usr/bin/python /usr/bin/salt-master MWorker-\n\nJul 21 11:37:45 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:38:45 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:39:46 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:40:46 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:41:47 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:42:47 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:43:48 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:44:48 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:45:49 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\nJul 21 11:46:50 localhost salt-master[1296]: [WARNING ] Cannot resolve address None info via socket: <class 'socket.gaierror'>\n"

},

{

"id": 2,

"name": "opencdp-systemd",

"status": "Unknown",

"usage": "\* opencdp-systemd.service - opencdp System Exec Daemon\n Loaded: loaded (/usr/lib/systemd/system/opencdp-systemd.service; disabled; vendor preset: disabled)\n Active: inactive (dead)\n"

},

{

"id": 3,

"name": "opencdp-worker",

"status": "Unknown",

"usage": "\* opencdp-worker.service - opencdp System worker Daemon\n Loaded: loaded (/usr/lib/systemd/system/opencdp-worker.service; disabled; vendor preset: disabled)\n Active: inactive (dead)\n"

},

{

"id": 4,

"name": "firewalld",

"status": "Unknown",

"usage": "\* firewalld.service - firewalld - dynamic firewall daemon\n Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)\n Active: inactive (dead)\n"

},

{

"id": 5,

"name": "opencdp-schedule",

"status": "Unknown",

"usage": "\* opencdp-schedule.service - opencdp System Schedule Daemon\n Loaded: loaded (/usr/lib/systemd/system/opencdp-schedule.service; disabled; vendor preset: disabled)\n Active: inactive (dead)\n"

},

{

"id": 7,

"name": "firewalld",

"status": "stop",

"usage": "\* firewalld.service - firewalld - dynamic firewall daemon\n Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)\n Active: inactive (dead)\n"

},

{

"id": 8,

"name": "firewalld",

"status": "stop",

"usage": "\* firewalld.service - firewalld - dynamic firewall daemon\n Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)\n Active: inactive (dead)\n"

}

]

}

## 设置某个服务

POST

{

"name": "firewalld",

"status": "stop"

}

# 审计系统

察看用户操作

## 察看审计信息

<http://192.168.50.23:8000/api/auditlogs>

GET

{

"count": 6,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/auditlogs/16>",

"id": 16,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "destroy",

"user": "test",

"starttime": "2017-09-20T08:53:26Z",

"endtime": "2017-09-20T09:00:17Z",

"exitcode": 204,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{}",

"urlpath": null

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/17>",

"id": 17,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "create",

"user": "test",

"starttime": "2017-09-20T09:48:04Z",

"endtime": "2017-09-20T09:48:04Z",

"exitcode": 201,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{\"comment\": \"test\", \"path\": \"/media/test/fu3\", \"csrfmiddlewaretoken\": \"dAD4eg8NPqeZKXb3nLkwtrvlNSfmXvAd\", \"name\": \"test\", \"volume\": \"http://192.168.50.23:8000/api/volumes/18\"}",

"urlpath": null

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/18>",

"id": 18,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "destroy",

"user": "test",

"starttime": "2017-09-20T09:49:22Z",

"endtime": "2017-09-20T09:59:43Z",

"exitcode": 204,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{}",

"urlpath": null

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/19>",

"id": 19,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "create",

"user": "test",

"starttime": "2017-09-20T10:09:27Z",

"endtime": "2017-09-20T10:09:27Z",

"exitcode": 201,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{\"comment\": \"test\", \"path\": \"/media/test/testfs\", \"csrfmiddlewaretoken\": \"ALl3MMKl2ofm99OlkhflyXPaWzAfrj9e\", \"name\": \"testv\", \"volume\": \"http://192.168.50.23:8000/api/volumes/18\"}",

"urlpath": "/api/rsyncs"

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/20>",

"id": 20,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "destroy",

"user": "test",

"starttime": "2017-09-20T10:15:04Z",

"endtime": "2017-09-20T10:15:04Z",

"exitcode": 204,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{}",

"urlpath": "/api/rsyncs/4"

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/21>",

"id": 21,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "create",

"user": "test",

"starttime": "2017-09-20T10:34:35Z",

"endtime": "2017-09-20T10:34:35Z",

"exitcode": 500,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{\"comment\": \"df\", \"path\": \"/tmp\", \"csrfmiddlewaretoken\": \"9KQIDlJpj710tNLg2D2iPRtY4tXCgKxO\", \"name\": \"testv\", \"volume\": \"http://192.168.50.23:8000/api/volumes/16\"}",

"urlpath": "/api/rsyncs"

}

]

}

## 查询审计可以根据时间,返回值组合查找

start\_datatime=

end\_datatime=

exit\_code=

<http://192.168.50.23:8000/api/auditlogs?exitcode=201&start_datetime=2017-09-20+09%3A00%3A00&end_datetime=2017-09-21>

GET

{

"count": 2,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.23:8000/api/auditlogs/17>",

"id": 17,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "create",

"user": "test",

"starttime": "2017-09-20T09:48:04Z",

"endtime": "2017-09-20T09:48:04Z",

"exitcode": 201,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{\"comment\": \"test\", \"path\": \"/media/test/fu3\", \"csrfmiddlewaretoken\": \"dAD4eg8NPqeZKXb3nLkwtrvlNSfmXvAd\", \"name\": \"test\", \"volume\": \"http://192.168.50.23:8000/api/volumes/18\"}",

"urlpath": null

},

{

"url": "<http://192.168.50.23:8000/api/auditlogs/19>",

"id": 19,

"host": "<http://192.168.50.23:8000/api/hosts/1>",

"command": "create",

"user": "test",

"starttime": "2017-09-20T10:09:27Z",

"endtime": "2017-09-20T10:09:27Z",

"exitcode": 201,

"module": "rsync.restapi",

"kwargs": "{}",

"request": "{\"comment\": \"test\", \"path\": \"/media/test/testfs\", \"csrfmiddlewaretoken\": \"ALl3MMKl2ofm99OlkhflyXPaWzAfrj9e\", \"name\": \"testv\", \"volume\": \"http://192.168.50.23:8000/api/volumes/18\"}",

"urlpath": "/api/rsyncs"

}

]

}

## 删除审计日志

删除审计日志需要用户有权限

"cmdlog.delete\_auditentry"

### 全部删除

<http://192.168.50.23:8000/api/auditlogs>

DELETE

### 单条记录删除

<http://192.168.50.23:8000/api/auditlogs/16>

DELETE

### 支持组合条件删除

<http://192.168.50.23:8000/api/auditlogs?exitcode=201&start_datetime=2017-09-20+09%3A00%3A00&end_datetime=2017-09-21>

DELETE

# 注册服务

## 获取服务

<http://192.168.50.62:8000/api/register>

GET

{

"count": 1,

"next": null,

"previous": null,

"results": [

{

"url": "<http://192.168.50.62:8000/api/register/14>",

"id": 14,

"registercode": "{\"validtime\": \"30\", \"company\": \"Heartsone\", \"endtime\": \"2017-10-21\", \"product\": \"CDP\", \"starttime\": \"2017-09-20\"}"

}

]

}

## 注册服务

POST

{

"registercode": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",

}