## 首页：

###### 获取全部虚拟机的平均CPU使用率

action: get

url: /all?metric=cpu\_idle&style=pie

respone: {"metric":"cpu\_idle","percent":val}

###### 获取全部虚拟机的平均负载

action: get

url: /all?metric=load\_five&style=pie

respone: {"metric":"load\_five","percent":val}

###### 获取全部虚拟机的平均内存使用率

action: get

url: /all?metric=mem\_usage&style=pie

respone: {"metric":"mem\_usage","percent":val}

###### 获取全部虚拟机的平均网络流量

action: get

url: /all?metric=bytes\_in

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /all?metric=bytes\_out

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 获取全部虚拟机的平均活跃进程

action: get

url: /all?metric=proc\_run

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /all?metric=proc\_total

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 获取全部虚拟机的平均磁盘使用率

action: get

url: /all?metric=reads

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /all?metric=writes

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 获取核心链路流量

action: post

url: /all/top/

body:{

"target":"管理节点ip",

"metric":"mem\_free",

"timestamp":"1day"

}

response:{"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 获取总虚拟机数量

action: get

url: /all/vms/

response: { "num":val }

###### 获取总物理机数量

action: get

url: /all/pcs/

response: { "num":val }

###### 获取云硬盘的总量

action: get

url: /all/nbs/

response: { "capacity":val }

###### 获取云平台用户数量

action: get

url: /all/users/

response: { "num":val }

## 用户：

###### 获取所有云平台用户(db1的CM\_UserTenant表）

**Action**

**GET**

**Url**

**GET** /data/users/

**Response**:

**HTTP 200 OK**

**Content-Type:** application/json

**Vary:** Accept

**Allow:** GET, HEAD, OPTIONS

[

{

"id": 1405042412157000000,

"username": "test1",

"tenantid": "1eacf8d1de7e4c7b9c0082d358ce66bb",

"usertype": 0,

"createtime": 1405042412174,

"email": "[test@debian.com](mailto:test@debian.com)",

"description": ""

}

]

###### 获取单个云平台用户详情(db1的CM\_UserTenant表）

**Action**

**GET**

**Url**

**/data/users/<id>/**

**GET** /data/users/1405042412157000000/

**Response**:

**HTTP 200 OK**

**Content-Type:** application/json

**Vary:** Accept

**Allow:** GET, HEAD, OPTIONS

{

"id": 1405042412157000000,

"username": "test1",

"tenantid": "1eacf8d1de7e4c7b9c0082d358ce66bb",

"usertype": 0,

"createtime": 1405042412174,

"email": "[test@debian.com](mailto:test@debian.com)",

"description": ""

}

###### 获取云平台用户总数(db1的CM\_UserTenant表）

**Action**

**GET**

**Url**

**GET** /data/users/count/

**Response**:

**HTTP 200 OK**

**Content-Type:** application/json

**Vary:** Accept

**Allow:** GET, HEAD, OPTIONS

{

"user\_count": 4

}

###### 获取云平台用户配额信息(db1的CM\_UserTenant表）

**Action**

**GET**

**Url**

**GET** /data/users/1405042412157000000/qutoas/

**Response**:

**HTTP 200 OK**

**Content-Type:** application/json

**Vary:** Accept

**Allow:** GET, HEAD, OPTIONS

{

"ecus": 29,

"private\_floating\_ips": 6,

"ram": 16384,

"floating\_ips": 5,

"floating\_ips\_limit": 5,

"instances": 8,

"cores": 14,

"local\_gb": 190,

"fixed\_ips": 8,

"fixed\_ips\_limit": 50

}

###### 2，获取所有用户的平均负载(按实际需求返回，如需要可返回"load\_fifteen","load\_one")

action: get

url: /users?metric="load\_five"

response:[{"metric":"load\_five","percent":val},...]

###### 3，获取所有用户的平均CPU使用率(按实际需求返回，如需要可返回"cpu\_user","cpu\_system")

action: get

url: /users?metric="cpu\_idle"

response:{"metric":"cpu\_idle","percent":val}

###### 4,获取所用用户的虚拟机数量

action:get

url:/users/vmnum/

response:[{ "user\_id":val,"user\_name":"xxx","num":val },{ "user\_id":val,"user\_name":"xxx","num":val }]

## 用户1：

###### 1， 获取全部虚拟机的平均CPU使用率

action: get

url: /user/{user\_id}/data?metric=cpu\_idle&style=line

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /user/{user\_id}/data?metric=cpu\_system&style=line

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 2， 获取全部虚拟机的平均负载

action: get

url:/user/{user\_id}/data?metric=load\_five&style=line

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url:/user/{user\_id}/data?metric=load\_fifth&style=line

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 3， 获取全部虚拟机的平均内存使用率

action: get

url: /user/{user\_id}/data?metric=mem\_usage&style=line

respone: {"metric":"mem\_usage","percent":val}

###### 4， 获取全部虚拟机的平均网络流量

action: get

url: /user/{user\_id}/data?metric=bytes\_in

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /user/{user\_id}/data?metric=bytes\_out

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 5， 获取全部虚拟机的平均活跃进程

action: get

url: /user/{user\_id}/data?metric=proc\_run

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /user/{user\_id}/data?metric=proc\_total

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 6， 获取全部虚拟机的平均磁盘使用率

action: get

url: /user/{user\_id}/data?metric=reads

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

action: get

url: /user/{user\_id}/data?metric=writes

respone: {"times":[xx,xx,xx],"values":[val1,val2,val3]}

###### 获取指定用户配额（nova）

action: get

url: /user/{user\_id}/usage/

response: [{"name":"instance", "usage":val},{"name":"ecu","usage"：val},...]

###### 获取指定用户的虚拟机列表

action: get

url: /users/{user\_id}/vmlist/

response:[{"uuid":"xxx","server\_name":"xxx","fixed\_ip":"xx.xx.xx.xx","floating\_ip":"xx.xx.xx.xx","az":"xxx","flavor":"xxxxx","image":"xxx"},...]

## 虚拟机详细信息：

###### 获取指定虚拟机监控数据

action: post

url: /getdata/

body:{

"monitor\_type": "vm",

"target":"xx.xx.xx.xx",

"metric":"mem\_free",

"timestamp":"1day"

}

response:{"times":[xx,xx,xx],"values":[val1,val2,val3]}

备注：target为虚拟机的fixed\_ip地址,metic值可以为（cpu\_idle,load\_five,mem\_free,mem\_totol,proc\_total,reads,writes, bytes\_in,bytes\_out,disk\_free,disk\_total, interrupts）,

Timestamp值：1hour,1day,1week,1mon,1year

###### 云主机信息

action: get

url: /vm/{fixed\_ip}/

response:{"uuid":"xxx","server\_name":"xxx","fixed\_ip":"xx.xx.xx.xx","floating\_ip":"xx.xx.xx.xx","az":"xxx","flavor":"xxxxx","image":"xxx","security\_group":"xxx",

"machine\_status":"xxx","power\_status":"xxx","keys":"xxx"}

## 主机：

###### 获取所有宿主机详细信息（待定）

action: get

url: /machine/cloud/

response:[{"uuid":"xxx","server\_name":"xxx","fixed\_ip":"xx.xx.xx.xx",},...]

###### 获取所有云平台外的物理机详细信息（待定）

action: get

url: /machine/outer/

response: :[{"uuid":"xxx","server\_name":"xxx","fixed\_ip":"xx.xx.xx.xx",},...]

###### 获取宿主机包括的虚拟机数量

action: get

url: /machine/vms/

response: { "num":val }

###### 获取宿主机平均负载

action: get

url:/ machine/{machine\_id}/data?metric=load\_five&style=pie

respone: {" metric ":” load\_five”,"values":val }

###### 获取宿主机平均ECU资源使用情况

action: get

url:/ machine/data?metric=ECU

respone: [{" hostname ":” xxx”,"totol":val,”used”:val },…,]

###### 获取宿主机平均mem资源使用情况

action: get

url:/ machine/data?metric=mem

respone: [{" hostname ":” xxx”,"totol":val,”used”:val },…,]

###### 获取宿主机平均云硬盘资源使用情况

action: get

url:/machine/data?metric=vol

respone: [{" hostname ":” xxx”,"totol":val,”used”:val },…,]

###### 获取宿主机平均实例存储资源使用情况

action: get

url:/ machine/data?metric=instance\_vol

respone: [{" hostname ":” xxx”,"totol":val,”used”:val },…,]

## 主机1

###### 获取指定物理机监控数据

action: post

url: /machine/{machine\_id}/data?metric=load\_five&style=line

body:{

"monitor\_type": "vm",

"target":"xx.xx.xx.xx",

"metric":"mem\_free",

"timestamp":"1day"

}

response:{"times":[xx,xx,xx],"values":[val1,val2,val3]}

备注：target为虚拟机的fixed\_ip地址,metic值可以为（cpu\_idle,load\_five,mem\_free,mem\_totol,proc\_total,reads,writes, bytes\_in,bytes\_out,disk\_free,disk\_total, interrupts）,

Timestamp值：1hour,1day,1week,1mon,1year

###### 获取指定物理机的主机信息

action: get

url: /machine/{machine\_id}/detail

respone: [{"hostname":” xxx”,"CPU":”val”,”mem”:”val”,”storage”:”xx”,”os”:”xx”,”starttime”:”xx”,”runtime”:”xx” },…,]

###### 获取指定用户的虚拟机列表

action: get

url: / machine /{ machine \_id}/vmlist/

response:[{"uuid":"xxx","server\_name":"xxx","fixed\_ip":"xx.xx.xx.xx","floating\_ip":"xx.xx.xx.xx","az":"xxx","flavor":"xxxxx","image":"xxx"},...]

## Vmware（待定）

## 报警

###### 获取报警历史

action: get

url: /alarm/history/

respone: [{"time":” xxx”,"metric\_rule":”val”,”object”:”val”,”alarm\_group”:”xx”,”detail”:”xx” },…,]

###### 获取报警统计

action: get

url: /alarm/count/

respone: {"alarm\_project":” xxx”,”alarm\_bind":”val” ,”alarm\_group”:”xx”,”alarm\_user”:”xx” }

###### 获取报警次数统计

action: get

url: /alarm/history/aggr

respone: {"day":[xx,xx,xx,…],”count”:[val,val,val,…] }

## 报警人员设置

###### 获取报警组(OK)

action: get

url: /alarm/groups/

respone: [{"pk”:val, ”**alarm\_group\_name**”:”xxx”,”**alarm\_group\_des**”:”xxx” },…]

###### 创建报警组(OK)

action: post

url: /alarm/group\_users/

body: {

"alarm\_group":{"alarm\_group\_name":"yyy","alarm\_group\_desc":"xxx"},

“alarm\_group\_user\_mapping":

[{"alarm\_user\_name":"user2",

"alarm\_user\_id":2,

"alarm\_config":1}…]

}

Respone status:

201 Created: 插入成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 修改报警组(OK)

action: update

url: /alarm/group\_user/{group\_id}/

body: {

"alarm\_group":{"alarm\_group\_name":"yyy","alarm\_group\_desc":"xxx"},

“alarm\_group\_user\_mapping":

[{"alarm\_user\_name":"user2",

"alarm\_user\_id":2,

"alarm\_config":1}…]

}

Respone status:

201 Created: 修改成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 删除报警组(OK)

action: delete

url: /alarm/group\_user/{group\_id}/

Respone status:

201 Created: 删除成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 获取某个报警组详细信息（OK）

action: get

url: /alarm/group\_users /{alarm\_group\_id}/

respone:

{

"alarm\_group": {

"pk": 37,

"alarm\_group\_name": "g1",

"alarm\_group\_desc": "xxx"

},

"alarm\_group\_user\_mapping": [

{

"alarm\_user\_id": 2,

"alarm\_group\_id": 37,

"alarm\_group\_name": "g1",

"alarm\_user\_name": "user2",

"alarm\_config": 1

},

{

"alarm\_user\_id": 4,

"alarm\_group\_id": 37,

"alarm\_group\_name": "g1",

"alarm\_user\_name": "user3",

"alarm\_config": 1

}

],

"alarm\_users": [

{

"pk": 2,

"alarm\_user\_name": "user2",

"email": "xxxx3",

"phone": "19822234412",

"qq": "312312321"

},

{

"pk": 4,

"alarm\_user\_name": "user3",

"email": "xxxx3",

"phone": "19822234412",

"qq": "312312321"

}

]

}

###### 获取全部报警组人员列表(OK)

action: get

url: /alarm/ users /

respone: [

{

"pk":2,

"alarm\_user\_name":"user2",

"email":"xxxx3",

"phone":"19822234412",

"qq":"312312321",

“remark”:”xxxx”

},

{

"pk":3,

"alarm\_user\_name":"user1",

"email":"xxxx3",

"phone":"19822234412",

"qq":"312312321",

“remark”:”xxxx”

}]

###### 创建报警组人员(OK)

action: post

url: /alarm/users/

body:{

" alarm\_user\_name":"xxx",

" email":”val”,

"phone":"xx ",

"qq”:”xxx”,

“remark”:”xxx”

}

Respone status:

201 Created: 创建成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 获取指定报警组人员信息(OK)

action: get

url: /alarm/user/3/

respone: {

"pk":3,

"alarm\_user\_name":"user1",

"email":"xxxx3",

"phone":"19822234412",

"qq":"312312321",

“remark”:”xxxx”

}

###### 修改报警组人员(OK)

action: update

url: /alarm/user/{user\_id}

body:{

" email":”val”,

"phone":"xx ",

"qq”:”xxx”,

“remark”:”xxx”

}

Respone status:

201 Created: 修改成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 删除报警组人员(OK)

action: delete

url: /alarm/user/{user\_id}

Respone status:

201 Created: 删除成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

## 报警规则设置

###### 获取报警项目列表

action: get

url: /alarm/project/

respone: [{"id”:val, ”alarm\_project\_name”:”xxx”, ”metic”:”xxx”, ”fixed\_ip”:”xxx”,” alarm\_rule\_name”:”xxx” },…]

###### 获取指定报警项目详细(待定)

action: get

url: /alarm/project/{project\_id}/

respone: {"id”:val, ”alarm\_project\_name”:”xxx”, ”alarm\_project\_desc”:”xxx”, ”fixed\_ip”:”xxx” ,” alarm\_group\_name”:”xx”, ”create\_time”:”xxx”, “metric\_rule”:[ {”metric\_name”:”xxx”, ”metric\_exp”:”xxx”,” alarm\_times”:val },…] }

###### 创建报警项目

action: post

url: /alarm/ project /

body:{

" alarm\_project\_name": "xxx",

" alarm\_project\_desc ":"xxx",

"alarm\_group\_id":val,

" fixed\_ip ":"xx",

“metric\_rule”:[ {”metric\_name”:”xxx”, ”metric\_exp”:”xxx”,” alarm\_times”:val },…]

}

Respone status:

201 Created: 插入成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 修改报警项目(报警规则先删除，再添加)

action: update

url: /alarm/ project /{project\_id}

body:{

" alarm\_project\_name": "xxx",

" alarm\_project\_desc ":"xxx",

"alarm\_group\_id":val,

" fixed\_ip ":"xx",

“metric\_rule”:[ {”metric\_name”:”xxx”, ”metric\_exp”:”xxx”,” alarm\_times”:val },…]

}

Respone status:

201 Created: 修改成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 删除报警项目(报警规则先删除，再添加)

action: delete

url: /alarm/project/{project\_id}

Respone status:

201 Created: 修改成功

400 Bad Request: 请求的内容包含不可更新属性

500 INTERNAL\_SERVER\_ERROR：内部错误

###### 获取报警组

action: get

url: /alarm/group/

respone: [{"id”:val,”name”:”xxx”,”object”:”xxx” },…]

###### 获取报警对象

action: get

url: /alarm/object/

respone: [{"fixed\_ip”:”xxx”,”host\_name”:”xxx”},…]

###### 获取报警组的报警规则

action: get

url: /alarm/rule/{project\_id}

respone: “metric\_rule”:[ {”metric\_name”:”xxx”, ”metric\_exp”:”xxx”,” alarm\_times”:val },…]