

# Billing Statistics

## API Essentials

## API Key Generation

## Autoscaling

- Add Autoscaling Rules
- Get List of Autoscaling Rules for VS
- Remove Autoscaling Rules

## Backups

- Add/Edit Note
- Convert Backup to Template
- Create Backup
- Delete Backup
- Get List of All VS Backups
- Get List of Incremental Backups
- Get List of Normal Backups
- Restore Disk from Backup

## Credit

- Get Credit

## DNS

- Add DNS Record
- Add DNS Zone
- Delete DNS Record
- Delete DNS Zone
- Edit DNS Record
- Get DNS Zones
- Get List of DNS Zone Records
- Get List of Name Servers

## Firewall Rules

- Add Firewall Rule
- Get Firewall Rules
- Update Firewall Rules

## Flex Cloud VM Control

## Getting Started with Superb Flex Cloud

## IP Addresses

- Get IP Address Joins

## Logs

- Get List of Log Items
- Get List of Transactions
- Get list of VS Transaction

## Network Interfaces

## GET /virtual\_machines/:virtual\_machine\_id/vm\_stats

Returns Billing Statistics of the specified VM

### Sections:

- Response Parameters
- JSON Request example
- JSON Response
- XML Request example
- XML Response example

### Response Parameters

*created\_at* – the timestamp in DB when this record was created

*currency\_code* - currency in which this virtual machine is charged within the billing plan

*id* – the ID of the server hourly statistics. You can add this parameter to the request URL to get a shorter statistics output.

*stat\_time* – the particular hour for which these statistics were generated

*updated\_at* – the date when these statistics were updated

*user\_id* - the ID of VS owner

*virtual\_machine\_id* - the ID of the VS

*vm\_billing\_stat\_id* - billing statistics ID

*billing\_stats* - an array of billing details for the resources used by this VM

*disks* - an array of disks used by this VM with their billing details:

*id* - disk ID used in database

*costs*- an array of disk related resources with their total prices for the period specified in the *stat-time* parameter, where:

*value* - the amount of resources used (GBs of disk size, Kbs of data read/written, the number of reads/writes)

*cost* - the total due for the resource

*resource\_name* - the resource in question. This can be *disk\_size*, *data\_read*, *data\_written*, *reads\_completed* and *writes\_completed*

*label* - disk name used in UI

*network\_interfaces* - an array of network interfaces used by this VM with their billing statistics:

*id* - network interface ID

*costs*- an array of network interface related resources with their total prices for the period specified in the *stat-time* parameter, where:

*value* - the amount of resources used by this network interface (the number of IPs, the port speed in Mb per second, the data sent and received in Gbs )

*cost* - the total due for the resource

*resource\_name*- the resource in question. This can be *ip\_addresses*, *rate*, *data\_received* and *data\_sent*

*label* - network interface name

*virtual\_machine* - an array of virtual machine billing details:

*id* - virtual server ID

- Get VS Network Interfaces
- Rebuild VS Network

## Recipes

- Add Recipe
- Add Recipe Step
- Assign Recipe to Virtual Server
- Delete Recipe
- Delete Recipe Step
- Edit Recipe
- Edit Recipe Step
- Get All Recipes
- Get Recipe Steps
- Get Virtual Server Recipes
- Remove recipe from Virtual Server
- Run Recipe on Multiple Virtual Servers
- Swap Recipe Step Number

## SSH Keys

- Add SSH Key
- Delete SSH Key
- Edit SSH Key
- Get SSH Keys
- Set SSH Keys on VS

## Templates

- Get Templates

## Test Route

## Troubleshooting API Issues

## Viewing Activity Logs

## Virtual Server Operating Systems

### Virtual Servers

- Add Virtual Server
- Billing Statistics
- Build Virtual Server
- Delete Virtual Server
- Edit Virtual Server
- Get CPU Usage Statistics
- Get List of Virtual Machines
  - Get specific VM Details
  - Search Virtual Servers by label
- Get statuses for All VMs
  - Get Specific VS Status
- Reboot Virtual Server
- Reset VS Password
- Shutdown Virtual Server
- Startup a Virtual Server
- Stop Virtual Server

*costs*- an array of VS resources with their total prices for the period specified in the *stat-time* parameter, where:

*value* - the amount of resources allocated to this VM. For the templates resource, this parameter means a template ID in database.

*cost* - the total due for this resource

*resource\_name* - the resource in question. This can be *cpu\_shares*, *cpus*, *memory*, *cpu\_usage* and *template*

*label* - VS name

*total\_cost* – the total amount of money owed for the VM specified by *id* parameter for a particular hour specified by *stat\_time* parameter ( $total\_cost = vm\_resources\_cost + usage\_cost$ )

*vm\_resources\_cost* – the amount of money due for the VM resources for the particular hour specified by *stat\_time* parameter (memory, disks, templates)

*usage\_cost* – the total due for VM usage for this particular hour specified by *stat\_time* parameter (data sent/received, bandwidth, CPU usage)

## JSON Request example

```
curl -i -X GET -H 'Accept: application/json' -H
'Content-type: application/json' -u
user:userpass --url
<api_url>/virtual_machines/:virtual_machine_id/
vm_stats.json?period[startdate]=YYYY-MM-DD+hh:m
m:ss&period[enddate]=YYYY-MM-DD+hh:mm:ss&period
[use_local_time]=1
```

## JSON Response

```
[
  {
    "vm_hourly_stat": {
      "created_at":
"2015-08-26T21:00:18Z",
      "currency_code": "USD",
      "id": 620994,
      "stat_time":
"2015-08-26T21:00:00Z",
      "updated_at":
"2015-08-26T21:00:18Z",
      "user_id": 337,
      "virtual_machine_id": 509,
      "vm_billing_stat_id": 4051,
      "billing_stats": {
        "disks": [
          {
            "id": 830,
            "costs": [
              {
                "value": 5,
                "cost":
```

## VS Disks

- Add New Disk
- Delete Disk
- Edit Disk
- Get VS Disks

```
0.00138889998197556,
  "resource_name": "disk_size"
    },
    {
      "value":
555874,
      "cost": 0,
  "resource_name": "data_read"
    },
    {
      "value": 79240,
      "cost": 0,
  "resource_name": "data_written"
    },
    {
      "value": 23052,
      "cost": 0,
  "resource_name": "reads_completed"
    },
    {
      "value": 1940,
      "cost": 0,
  "resource_name": "writes_completed"
    }
  ],
  "label": "Disk#830"
},
{
  "id": 831,
  "costs": [
    {
      "value": 1,
      "cost":
0.000277779996395111,
  "resource_name": "disk_size"
    },
    {
      "value": 5164,
      "cost": 0,
  "resource_name": "data_read"
    },
    {
      "value": 0,
      "cost": 0,
  "resource_name": "data_written"
```

```
        },
        {
            "value": 1298,
            "cost": 0,
        },
        "resource_name": "reads_completed"
    },
    {
        "value": 0,
        "cost": 0,
    },
    "resource_name": "writes_completed"
},
"label": "Disk#831"
}
],
"network_interfaces": [
    {
        "id": 507,
        "costs": [
            {
                "value": 1,
                "cost": 0,
            },
            "resource_name": "ip_addresses"
        ],
        {
            "value": 1,
            "cost": 0,
        },
        "resource_name": "rate"
    },
    {
        "value": 1171,
        "cost": 0,
    },
    "resource_name": "data_received"
},
{
    "value": 26,
    "cost":
0.00000123977656585339,
"resource_name": "data_sent"
},
"label": "eth0"
}
],
"virtual_machines": [
    {
        "id": 509,
```

```
        "costs": [
            {
                "value": 1,
                "cost":
0.000694439979270101,
                "resource_name": "cpu_shares"
            },
            {
                "value": 384,
                "cost":
0.00533375982195139,
                "resource_name": "memory"
            },
            {
                "value": 128,
                "cost": 0,
                "resource_name": "template"
            },
            {
                "value": 5,
                "cost": 0,
                "resource_name": "cpu_usage"
            }
        ],
        "label": "ZIGGY"
    }
]
},
"total_cost": 0.00769611978,
"vm_resources_cost": 0.00769488,
"usage_cost": 0.00000123978,
"booted": true
```

```
}  
}  
]
```

### XML Request example

```
curl -i -X GET -H 'Accept: application/json' -H  
'Content-type: application/json' -u  
user:userpass --url  
<api_url>/virtual_machines/:virtual_machine_id/  
vm_stats.xml?period[startdate]=YYYY-MM-DD+hh:mm  
:ss&period[enddate]=YYYY-MM-DD+hh:mm:ss&period[  
use_local_time]=1
```

### XML Response example

```
<?xml version="1.0" encoding="UTF-8"?>  
<vm_hourly_stats type="array">  
<vm_hourly_stat>  
<created_at  
type="datetime">2011-08-09T12:00:10Z</created_a  
t>  
<currency_code>USD</currency_code>  
<id type="integer">8248</id>  
<stat_time  
type="datetime">2011-08-09T12:00:00Z</stat_time  
>  
<updated_at  
type="datetime">2011-08-09T12:00:10Z</updated_a  
t>  
<user_id type="integer">1</user_id>  
<virtual_machine_id  
type="integer">44</virtual_machine_id>  
<vm_billing_stat_id  
type="integer">100175</vm_billing_stat_id>  
<billing_stats>  
<disks type="array">  
<disk>  
<id type="integer">2933</id>  
<costs type="array">  
<cost>  
<value type="integer">5</value>  
<cost type="float">3.0</cost>  
<resource_name>disk_size</resource_name>  
</cost>  
</costs>  
<label>Disk#2933</label>  
</disk>
```

```
<disk>...</disk>
</disks>
<network_interfaces type="array">
  <network_interface>
    <id type="integer">2688</id>
    <costs type="array">
      <cost>
        <value type="integer">1</value>
        <cost type="float">0.0</cost>
        <resource_name>ip_addresses</resource_name>
      </cost>
      <cost>...</cost>
    </costs>
    <label>eth0</label>
  </network_interface>
</network_interfaces>

<virtual_machines type="array">
  <virtual_machine>
    <id type="integer">1701</id>
    <costs type="array">
      <cost>
        <value type="integer">1</value>
        <cost type="float">0.0</cost>
        <resource_name>cpus</resource_name>
      </cost>
    </costs>
    <label>zaza_CP_3.2 (do not remove)</label>
  </virtual_machine>
</virtual_machines>
</billing_stats>
<total_cost type="float">0.0</total_cost>
<vm_resources_cost
type="float">0.0</vm_resources_cost>
<usage_cost type="float">0.0</usage_cost>
```

```
</vm_hourly_stat>  
<vm_hourly_stat>...</vm_hourly_stat>  
</vm_hourly_stats>
```