

Apache CloudStack 4.1.1

CloudStack 開發者指南

版



Apache CloudStack

法律聲明

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

摘要

這個指南教您如何開發CloudStack、使用API來操作及整合、存取資料及使用CloudStack特定工具來簡化開發、測試和整合

1. 概念

- 1.1. 甚麼是CloudStack ?
- 1.2. What Can CloudStack Do?
- 1.3. 架設架構總攬
 - 1.3.1. Management Server Overview
 - 1.3.2. 雲端基礎架構簡介
 - 1.3.3. 網路簡介

2. 使用 Maven 建立 CloudStack

- 2.1. 從原始碼建立 CloudStack
- 2.2. 建立程序事前準備
- 2.3. 建立步驟
- 2.4. 部署及測試步驟

3. Introduction to the CloudStack API

- 3.1. Roles
- 3.2. API Reference Documentation
- 3.3. Getting Started

4. 有甚麼新的API?

- 4.1. 4.1版本
 - 4.1.1. Reconfiguring Physical Networks in VMs
 - 4.1.2. IPv6 Support in CloudStack
 - 4.1.3. 額外的VMX設定
 - 4.1.4. 重新設定 SSH金鑰來存取VM
 - 4.1.5. Changed API Commands in 4.1
 - 4.1.6. 4.1-incubating新增API指令
- 4.2. 4.0版本
 - 4.2.1. 4.0.0-incubating改變API指令
 - 4.2.2. 4.0.0-incubating新增API指令
- 4.3. 3.0版本
 - 4.3.1. 啟用8096網路連接埠
 - 4.3.2. 已停止的虛擬機器
 - 4.3.3. Change to Behavior of List Commands
 - 4.3.4. Removed API commands
 - 4.3.5. Added API commands in 3.0
 - 4.3.6. 增加的CloudStack錯誤碼

5. 對CloudStack API 發出請求

- 5.1. Making API Requests
- 5.2. Signing API Requests
 - 5.2.1. How to sign an API call with Python
- 5.3. Enabling API Call Expiration
- 5.4. Limiting the Rate of API Requests

- 5.4.1. Configuring the API Request Rate
- 5.4.2. Limitations on API Throttling
- 5.5. Responses
 - 5.5.1. Response Formats: XML and JSON
 - 5.5.2. Maximum Result Pages Returned
 - 5.5.3. Error Handling
- 5.6. Asynchronous Commands
 - 5.6.1. 工作狀態
 - 5.6.2. Example
- 6. 使用使用資料
 - 6.1. Usage Record Format
 - 6.1.1. Virtual Machine Usage Record Format
 - 6.1.2. Network Usage Record Format
 - 6.1.3. IP位址使用紀錄的格式
 - 6.1.4. Disk Volume Usage Record Format
 - 6.1.5. 模組、ISO及快取物件的使用紀錄格式
 - 6.1.6. Load Balancer Policy 或 Port Forwarding Rule 的使用紀錄格式
 - 6.1.7. Network Offering Usage Record Format
 - 6.1.8. VPN使用者使用紀錄格式
 - 6.2. Usage Types
 - 6.3. Example response from listUsageRecords
 - 6.4. Dates in the Usage Record
 - 6.5. 廣域設定限制
- 7. 準備及建立CloudStack文件
 - 7.1. 安裝
 - 7.2. 建立CloudStack文件
 - 7.3. 撰寫CloudStack文件
 - 7.4. 翻譯CloudStack文件
 - 7.4.1. 翻譯CloudStack文件
- 8. 工具
 - 8.1. DevCloud
 - 8.1.1. DevCloud使用模式
 - 8.1.2. 建立DevCloud
 - 8.2. Marvin
 - 8.2.1. 建立及安裝 Marvin
 - 8.3. CloudMonkey
 - 8.3.1. Installing CloudMonkey
 - 8.3.2. 系統設定
 - 8.3.3. API Discovery
 - 8.3.4. Tabular Output
 - 8.3.5. Interactive Shell Usage
 - 8.3.6. Starting a Virtual Machine instance with CloudMonkey
 - 8.3.7. Scripting with CloudMonkey
 - 8.4. Apache Libcloud
- A. Event Types
- B. 警告
- C. 時區
- D. 修訂記錄

章 1. 概念

- 1.1. 甚麼是CloudStack ?
- 1.2. What Can CloudStack Do?
- 1.3. 架設架構總攬
 - 1.3.1. Management Server Overview
 - 1.3.2. 雲端基礎架構簡介
 - 1.3.3. 網路簡介

1.1. 甚麼是CloudStack ?

CloudStack 是一個開放原始碼的軟體，將運算資源抽象化成一整個資源庫提供了公有、私有以及混和式的雲端平台服務 (IaaS)。CloudStack具備了管理網路、儲存裝置和計算資源的能力，使用者可以運用CloudStack 部署、管理、設定雲端環境

一般使用者為服務提供者及企業，有了CloudStack，您可以：

- ▶ 依照需求建立一個具備彈性的雲端服務，網路服務提供者可以販售虛擬機、儲存服務、網路設定服務。
- ▶ 建立一個只提供內部員工所使用的私有雲服務，與傳統管理實體主機的方式有所不同，企業員工不需透過IT部門即可自助式的使用虛擬機





1.2. What Can CloudStack Do?

Multiple Hypervisor Support

CloudStack works with a variety of hypervisors, and a single cloud deployment can contain multiple hypervisor implementations. The current release of CloudStack supports pre-packaged enterprise solutions like Citrix XenServer and VMware vSphere, as well as KVM or Xen running on Ubuntu or CentOS.

Massively Scalable Infrastructure Management

CloudStack can manage tens of thousands of servers installed in multiple geographically distributed datacenters. The centralized management server scales linearly, eliminating the need for intermediate cluster-level management servers. No single component failure can cause cloud-wide outage. Periodic maintenance of the management server can be performed without affecting the functioning of virtual machines running in the cloud.

Automatic Configuration Management

CloudStack automatically configures each guest virtual machine's networking and storage settings.

CloudStack internally manages a pool of virtual appliances to support the cloud itself. These appliances offer services such as firewalling, routing, DHCP, VPN access, console proxy, storage access, and storage replication. The extensive use of virtual appliances simplifies the installation, configuration, and ongoing management of a cloud deployment.

Graphical User Interface

CloudStack offers an administrator's Web interface, used for provisioning and managing the cloud, as well as an end-user's Web interface, used for running VMs and managing VM templates. The UI can be customized to reflect the desired service provider or enterprise look and feel.

API and Extensibility

CloudStack provides an API that gives programmatic access to all the management features available in the UI. The API is maintained and documented. This API enables the creation of command line tools and new user interfaces to suit particular needs. See the Developer's Guide and API Reference, both available at [Apache CloudStack Guides](#) and [Apache CloudStack API Reference](#), respectively.

The CloudStack pluggable allocation architecture allows the creation of new types of allocators for the selection of storage and Hosts. See the [Allocator Implementation Guide](#) (http://docs.cloudstack.org/CloudStack_Documentation/Allocator_Implementation_Guide).

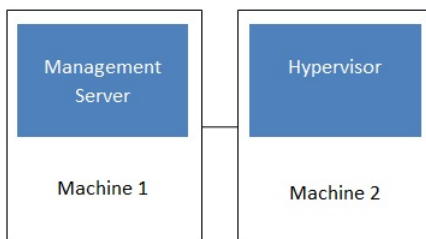
High Availability

CloudStack has a number of features to increase the availability of the system. The Management Server itself may be deployed in a multi-node installation where the servers are load balanced. MySQL may be configured to use replication to provide for a manual failover in the event of database loss. For the hosts, CloudStack supports NIC bonding and the use of separate networks for storage as well as iSCSI Multipath.

1.3. 架設架構總攬

CloudStack的安裝包含兩個部分：管理伺服器及雲端基礎架構。當您架設及管理一個CloudStack雲端時，您需要提供如主機、儲存裝置及IP位址等資源給管理伺服器，而管理伺服器幫您管理這些資源。

產品的最少安裝包含一個執行CloudStack管理伺服器的機器及雲端基礎架構的機器(這個例子是只包含一個執行超級監督者程式的主機)，在最小的架構中，一個機器可以同時執行管理伺服器及超級監督者主機(使用 KVM hypervisor)



Simplified view of a basic deployment

完全安裝包含多點管理伺服器及上千上萬使用多種進階網路設定的主機，更多資訊詳見 \$PRODUCT;安裝指南的"Choosing a Deployment Architecture"部分

1.3.1. Management Server Overview

The Management Server is the CloudStack software that manages cloud resources. By interacting with the Management Server through its UI or API, you can configure and manage your cloud infrastructure.

The Management Server runs on a dedicated server or VM. It controls allocation of virtual machines to hosts and assigns storage and IP addresses to the virtual machine instances. The Management Server runs in a Tomcat container and requires a MySQL database for persistence.

The machine must meet the system requirements described in System Requirements.

The Management Server:

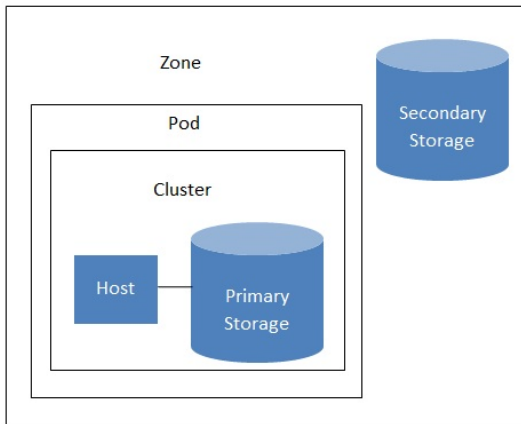
- » Provides the web user interface for the administrator and a reference user interface for end users.
- » Provides the APIs for CloudStack.
- » Manages the assignment of guest VMs to particular hosts.
- » Manages the assignment of public and private IP addresses to particular accounts.
- » Manages the allocation of storage to guests as virtual disks.
- » Manages snapshots, templates, and ISO images, possibly replicating them across data centers.
- » Provides a single point of configuration for the cloud.

1.3.2. 雲端基礎架構簡介

管理伺服器管理多個區域(通常為資料中心)，包含訪客虛擬機器的主機，雲端基礎架構可以組織為下：

- » Zone：通常，區域等於一個資料中心。區域包含一至多個pods極次要儲存裝置

- Pod：通常是一層硬體架構，包含 layer-2 交換器及一至多個叢集
- Cluster：通常包含一至多個主機及主要儲存裝置
- Host：叢集中的運算節點，以訪客虛擬機器的形式在實際的雲端運行
- 主要儲存裝置連接到叢集，存放所有 VM 的硬碟容量
- 次要儲存裝置連接到區域，儲存模組、ISO 映像及硬碟容量快取物件



Nested organization of a zone

More Information

更多資訊，請參閱 cloud infrastructure concepts 的文件

1.3.3. 網路簡介

CloudStack 提供兩種網路範本：

- 基本。為類似 AWS 模式的網路架構，提供 layer-3 的 Security group 安全機制 (IP 位置過濾機制)
- 進階。提供使用者更多的網路拓撲結構，選擇此選項將可更彈性的設定網路

更多細節，見 Network Setup

章 2. 使用 Maven 建立 CloudStack

2.1. 從原始碼建立 CloudStack

2.1. 建立程序事前準備

2.3. 建立步驟

2.4. 部署及測試步驟

2.1. 從原始碼建立 CloudStack

注意

Prior to the 4.0.0 incubating release, Ant was used to build CloudStack. A migration to Maven started in the 4.0.0 cycle, and has completed in 4.1.0.

網站及維基皆包含最新的資訊：

- <https://cwiki.apache.org/confluence/display/CLOUDSTACK/How+to+build+on+master+branch>
- <https://cwiki.apache.org/confluence/display/CLOUDSTACK/Setting+up+CloudStack+Development+Environment>

建立 CloudStack 的首要步驟為：

- 安裝先決條件及設定環境
- 理解 Maven 外型的多樣性及建立目標
- 部署及測試
- 如果需要，學習如何建立二進為文件

注意

學習 Maven 不再此文件範圍中

到 Maven 網站 <http://maven.apache.org/guides/getting-started/index.html>

2.2. 建立程序事前準備

此章節，我們假設您是使用 Ubuntu Linux distribution 的 Advanced Packaging Tool (APT)，如果您是使用不同的產品或作業系統及不同的 packaging tool，請依循以下指示來適合您的環境。建立 CloudStack 您需要：

- git, <http://git-scm.com>

```
sudo apt-get install git-core
```

- maven, <http://maven.apache.org>

```
sudo apt-get install maven
```

Make sure that you installed maven 3

```
$ mvn --version
Apache Maven 3.0.4
Maven home: /usr/share/maven
Java version: 1.6.0_24, vendor: Sun Microsystems Inc.
java home: /usr/lib/jvm/java-6-openjdk-amd64/jre
```

```
java home: /usr/lib/jvm/java-6-openjdk-amd64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.2.0-33-generic", arch: "amd64", family: "unix"
```

» java

設定 JAVA_HOME 環境變數

```
$ export JAVA_HOME=/usr/lib/jvm/java-6-openjdk
```

除此之外，想要驗開發環境下部署及執行 CloudStack，您需要：

» Mysql

```
sudo apt-get install mysql-server-5.5
```

啟動 mysqld 服務，並建立雲端使用者及密碼為 cloud

» Tomcat 6

```
sudo apt-get install tomcat6
```

2.3. 建立步驟

CloudStack 使用 git 來控制原始碼版本，請先拉它確定您有原始碼

```
git clone https://git-wip-us.apache.org/repos/asf/cloudstack.git
```

許多 Project Object Models (POM) 定義為處理多種 CloudStack 建立目標，特定功能需要以些不相容 Apache license 的 packages，因此需要自行下載，檢查 Wiki <https://cwiki.apache.org/CloudStack/building-with-maven.html>：

```
mvn clean install
```

結果的 jar 檔案會在目標資料夾的子資料夾下，為編譯過的組件

2.4. 部署及測試步驟

部署您編譯的 CloudStack code 有兩個步驟：

1. 如果您還沒設定資料庫或修改其性質，請執行：

```
mvn -P developer -pl developer -Ddeploydb
```

2. 然後，您需要執行 CloudStack 管理伺服器，連結一個除錯器到伺服器：

```
export MAVEN_OPTS="-Xmx1024 -Xdebug -Xrunjdwp:transport=dt_socket,address=8787,server=y,suspend=n"
```

```
mvn -pl :cloud-client-ui jetty:run
```



警告

處理資料庫時，記得，您可能整個刪掉它及失去所有資料中心設定

章 3. Introduction to the CloudStack API

3.1. Roles

3.2. API Reference Documentation

3.3. Getting Started

3.1. Roles

The CloudStack API supports three access roles:

1. Root Admin. Access to all features of the cloud, including both virtual and physical resource management.
2. Domain Admin. Access to only the virtual resources of the clouds that belong to the administrator's domain.
3. User. Access to only the features that allow management of the user's virtual instances, storage, and network.

3.2. API Reference Documentation

You can find all the API reference documentation at the below site:

<http://cloudstack.apache.org/docs/api/>

3.3. Getting Started

To get started using the CloudStack API, you should have the following:

- » URL of the CloudStack server you wish to integrate with.
- » Both the API Key and Secret Key for an account. This should have been generated by the administrator of the cloud instance and given to you.
- » Familiarity with HTTP GET/POST and query strings.
- » Knowledge of either XML or JSON.
- » Knowledge of a programming language that can generate HTTP requests; for example, Java or PHP.

章 4. 有甚麼新的 API?

4.1. 4.1 版本

4.1.1. Reconfiguring Physical Networks in VMs

4.1.2. IPv6 Support in CloudStack

4.1.3. 額外的 VM 設定

4.1.4. 重新設定 SSH 金鑰來存取 VM

4.1.5. Channed API Commands in 4.1

4.1.6. Changed API Commands in 4.1

4.1.6. 4.1-incubating新增API指令

4.2. 4.0版本

4.2.1. 4.0.0-incubating改變API指令

4.2.2. 4.0.0-incubating新增API指令

4.3. 3.0版本

4.3.1. 啟用8096網路連接埠

4.3.2. 已停止的虛擬機器

4.3.3. Change to Behavior of List Commands

4.3.4. Removed API commands

4.3.5. Added API commands in 3.0

4.3.6. 增加的CloudStack錯誤碼

以下描述每個CloudStack版本中任何新的主要功能，功能適用於API

4.1. 4.1版本

4.1.1. Reconfiguring Physical Networks in VMs

CloudStack provides the ability to move VMs between networks and reconfigure a VM's network. You can remove a VM from a physical network and add to a new physical network. You can also change the default physical network of a virtual machine. With this functionality, hybrid or traditional server loads can be accommodated with ease.

This feature is supported on XenServer and KVM hypervisors.

The following APIs have been added to support this feature. These API calls can function only while the VM is in running or stopped state.

4.1.1.1. addNicToVirtualMachine

The addNicToVirtualMachine API adds a new NIC to the specified VM on a selected network.

parameter	description	數值
virtualmachineid	The unique ID of the VM to which the NIC is to be added.	true
networkid	The unique ID of the network the NIC that you add should apply to.	true
ipaddress	The IP address of the VM on the network.	false

The network and VM must reside in the same zone. Two VMs with the same name cannot reside in the same network. Therefore, adding a second VM that duplicates a name on a network will fail.

4.1.1.2. removeNicFromVirtualMachine

The removeNicFromVirtualMachine API removes a NIC from the specified VM on a selected network.

parameter	description	數值
virtualmachineid	The unique ID of the VM from which the NIC is to be removed.	true
nicid	The unique ID of the NIC that you want to remove.	true

Removing the default NIC is not allowed.

4.1.1.3. updateDefaultNicForVirtualMachine

The updateDefaultNicForVirtualMachine API updates the specified NIC to be the default one for a selected VM.

parameter	description	數值
virtualmachineid	The unique ID of the VM for which you want to specify the default NIC.	true
nicid	The unique ID of the NIC that you want to set as the default one.	true

4.1.2. IPv6 Support in CloudStack

CloudStack supports Internet Protocol version 6 (IPv6), the recent version of the Internet Protocol (IP) that defines routing the network traffic. IPv6 uses a 128-bit address that exponentially expands the current address space that is available to the users. IPv6 addresses consist of eight groups of four hexadecimal digits separated by colons, for example, 5001:0d18:83a3:1012:1000:8s2e:0870:7454. CloudStack supports IPv6 for public IPs in shared networks. With IPv6 support, VMs in shared networks can obtain both IPv4 and IPv6 addresses from the DHCP server. You can deploy VMs either in a IPv6 or IPv4 network, or in a dual network environment. If IPv6 network is used, the VM generates a link-local IPv6 address by itself, and receives a stateful IPv6 address from the DHCPv6 server.

IPv6 is supported only on KVM and XenServer hypervisors. The IPv6 support is only an experimental feature.

Here's the sequence of events when IPv6 is used:

1. The administrator creates an IPv6 shared network in an advanced zone.
2. The user deploys a VM in an IPv6 shared network.
3. The user VM generates an IPv6 link local address by itself, and gets an IPv6 global or site local address through DHCPv6.

For information on API changes, see [節 4.1.5, "Changed API Commands in 4.1"](#).

4.1.2.1. Prerequisites and Guidelines

Consider the following:

- » CIDR size must be 64 for IPv6 networks.
- » The DHCP client of the guest VMs should support generating DUID based on Link-layer Address (DUID-LL). DUID-LL derives from the MAC address of guest VMs, and therefore the user VM can be identified by using DUID. See [Dynamic Host Configuration Protocol for IPv6](#) for more information.
- » The gateway of the guest network generates Router Advertisement and Response messages to Router Solicitation. The M (Managed Address Configuration) flag of Router Advertisement should enable stateful IP address configuration. Set the M flag to where the end nodes receive their IPv6 addresses from the DHCPv6 server as opposed to the router or switch.

注意

The M flag is the 1-bit Managed Address Configuration flag for Router Advertisement. When set, Dynamic Host Configuration Protocol (DHCPv6) is available for address configuration in addition to any IPs set by using stateless address auto-configuration.

- » Use the System VM template exclusively designed to support IPv6. Download the System VM template from <http://cloudstack.apache.org/systemvm/>

- ▶ The concept of Default Network applies to IPv6 networks. However, unlike IPv4 CloudStack does not control the routing information of IPv6 in shared network; the choice of Default Network will not affect the routing in the user VM.
- ▶ In a multiple shared network, the default route is set by the rack router, rather than the DHCP server, which is out of CloudStack control. Therefore, in order for the user VM to get only the default route from the default NIC, modify the configuration of the user VM, and set non-default NIC's `accept_ra` to 0 explicitly. The `accept_ra` parameter accepts Router Advertisements and auto-configure `/proc/sys/net/ipv6/conf/interface` with received data.

4.1.2.2. Limitations of IPv6 in CloudStack

The following are not yet supported:

1. Security groups
2. Userdata and metadata
3. Passwords

4.1.2.3. Guest VM Configuration for DHCPv6

For the guest VMs to get IPv6 address, run `dhclient` command manually on each of the VMs. Use `DUID-LL` to set up `dhclient`.

注意

The IPv6 address is lost when a VM is stopped and started. Therefore, use the same procedure to get an IPv6 address when a VM is stopped and started.

1. Set up `dhclient` by using `DUID-LL`.

Perform the following for DHCP Client 4.2 and above:

- a. Run the following command on the selected VM to get the `dhcpv6` offer from VR:

```
dhclient -6 -D LL <dev>
```

Perform the following for DHCP Client 4.1:

- a. Open the following to the `dhclient` configuration file:

```
vi /etc/dhcp/dhclient.conf
```

- b. Add the following to the `dhclient` configuration file:

```
send dhcp6.client-id = concat(00:03:00, hardware);
```

2. Get IPv6 address from DHCP server as part of the system or network restart.

Based on the operating systems, perform the following:

On CentOS 6.2:

- a. Open the Ethernet interface configuration file:

```
vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

The `ifcfg-eth0` file controls the first NIC in a system.

- b. Make the necessary configuration changes, as given below:

```
DEVICE=eth0
HWADDR=06:A0:F0:00:00:38
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=dhcp6
TYPE=Ethernet
USERCTL=no
PEERDNS=yes
IPV6INIT=yes
DHCPV6C=yes
```

- c. Open the following:

```
vi /etc/sysconfig/network
```

- d. Make the necessary configuration changes, as given below:

```
NETWORKING=yes
HOSTNAME=centos62mgmt.lab.vmops.com
NETWORKING_IPV6=yes
IPV6_AUTOCONF=no
```

On Ubuntu 12.10

- a. Open the following:

```
etc/network/interfaces:
```

- b. Make the necessary configuration changes, as given below:

```
iface eth0 inet6 dhcp
autoconf 0
accept_ra 1
```

4.1.3. 額外的VMX設定

VMX (.vmx)檔案是虛擬機器的主要設定檔，當新的VM建立時，作業系統、硬碟大小及網路等資訊都存在這個檔案裡。VM主動將所有設定變更寫入.vmx檔案內。通常此檔案會在VM的資料夾內。對於Windows Vista / Windows 7 / Windows Server 2008，預設位置為C:\Users\your_user_name>My Documents\Virtual Machines\virtual_machine_name>.vmx。對於linux,vmware-cmd -l列出所有登陸的VMX檔案的完整路徑。任何從ESX/ESXi的手動新增都會從vCenter Server database的入口複寫掉。因此，請您在編輯檔案之前，先將VM從vCenter server's inventory移除，等更新完後再登錄回去

CloudStack API中，`registerTemplate`支援通過一些VMX設定。支援的參數為`rootDiskController`，`nicAdapter`及鍵盤。除了這些已知的VMX參數，您現在可以使用在`registerTemplate` API call中的 `keyboard.typeomaticMinDelay`參數。這個參數控制在遠端主機的重複按鍵延遲，想知道更多內容，參見[keyboard.typeomaticMinDelay](#)

4.1.4. 重新設定 SSH金鑰來存取VM

使用 `resetSSHKeyForVirtualMachine` API來設定或重設指定到虛擬機器的SSH keypair，由於多了這個功能，忘掉或有危害的SSH keypair可以被換掉，使用者可以使用新的keypair存取VM。建立或註冊新的keypair，然後呼叫 `resetSSHKeyForVirtualMachine`

4.1.5. Changed API Commands in 4.1

API Commands	Description
<code>createNetworkOffering</code>	The following request parameters have been added: <ul style="list-style-type: none"> ▶ <code>isPersistent</code> ▶ <code>startip6</code> ▶ <code>endip6</code> ▶ <code>ip6gateway</code> ▶ <code>ip6cidr</code>

	<ul style="list-style-type: none"> ipocidr
listNetworkOfferings listNetworks	<p>The following request parameters have been added:</p> <ul style="list-style-type: none"> isPersistent This parameter determines if the network or network offering listed are persistent or not. ip6gateway ip6cidr
createVlanIpRange	<p>The following request parameters have been added:</p> <ul style="list-style-type: none"> startip6 endip6 ip6gateway ip6cidr
deployVirtualMachine	<p>The following parameter has been added: ip6Address. The following parameter is updated to accept the IPv6 address: iptonetworklist.</p>
CreateZoneCmd	<p>The following parameter have been added: ip6dns1, ip6dns2.</p>
listRouters listVirtualMachines	<p>For nic responses, the following fields have been added.</p> <ul style="list-style-type: none"> ip6address ip6gateway ip6cidr
listVlanIpRanges	<p>For nic responses, the following fields have been added.</p> <ul style="list-style-type: none"> startip6 endip6 ip6gateway ip6cidr
listRouters listZones	<p>For DomainRouter and DataCenter response, the following fields have been added.</p> <ul style="list-style-type: none"> ip6dns1 ip6dns2
addF5LoadBalancer configureNetscalerLoadBalancer addNetscalerLoadBalancer listF5LoadBalancers configureF5LoadBalancer listNetscalerLoadBalancers	<p>The following response parameter is removed: inline.</p>
listFirewallRules createFirewallRule	<p>The following request parameter is added: traffictype (optional).</p>
listUsageRecords	<p>The following response parameter is added: virtualsize.</p>
deleteIso	<p>The following request parameter is added: forced (optional).</p>
createStoragePool	<p>The following request parameters are made mandatory:</p> <ul style="list-style-type: none"> podid clusterid
createAccount	<p>The following new request parameters are added: accountid, userid</p>
createUser	<p>The following new request parameter is added: userid</p>
createDomain	<p>The following new request parameter is added: domainid</p>
listZones	<p>The following request parameters is added: securitygroupenabled</p>

4.1.6. 4.1-incubating新增API指令

- createEgressFirewallRules (在訪客網路建立egress firewall rule)
- deleteEgressFirewallRules (在訪客網路刪除egress firewall rule)
- listEgressFirewallRules (為訪客網路設定的egress firewall rules表單)
- resetSSHKeyForVirtualMachine (重新設定虛擬機器的SSHkey)
- addBaremetalHost (新增主機)
- addNicToVirtualMachine (在選定的網路新增NIC到指定的VM)
- removeNicFromVirtualMachine(在選定的VM刪除指定的NIC)
- updateDefaultNicForVirtualMachine (在選定的VM更新指定的NIC成為預設值)
- addRegion (登路一個Region到另一個Region)
- updateRegion (更新Region細節：ID, 名稱, 端點, 使用者 API Key, 及使用者Secret Key)
- removeRegion (從現在的Region移除一個Region)
- listRegions (列出所有 Regions. 用ID 或 Name過濾)
- getUser (此API 只能由管理者使用. 用API Key取得使用者細節)

4.2. 4.0版本

4.2.1. 4.0.0-incubating改變API指令

API指令	敘述
copyTemplate prepareTemplate registerTemplate updateTemplate createProject activateProject suspendProject	<p>表中指令有一個新的response欄位, 其他的沒有改變 新的response欄位: tags(*)</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>注意</p> <p>很多其他的指令也有新的參數tags(*), 這些指令被分開表列</p> </div>

updateProject listProjectAccounts createVolume migrateVolume attachVolume detachVolume uploadVolume createSecurityGroup registerIso copyIso updateIso createIpForwardingRule listIpForwardingRules createLoadBalancerRule updateLoadBalancerRule createSnapshot	
rebootVirtualMachine attachIso detachIso listLoadBalancerRuleInstances resetPasswordForVirtualMachine changeServiceForVirtualMachine recoverVirtualMachine startVirtualMachine migrateVirtualMachine deployVirtualMachine assignVirtualMachine updateVirtualMachine restoreVirtualMachine stopVirtualMachine destroyVirtualMachine	表中的指令有兩個新的response欄位，其他的沒有改變 新增response欄位：keypair, tags(*)
listSecurityGroups listFirewallRules listPortForwardingRules listSnapshots listIcos listProjects listTemplates listLoadBalancerRules	表中指令有新增以下欄位，其他的沒有變 新增request欄位：tags (選擇性) 新的response欄位: tags(*)
listF5LoadBalancerNetworks listNetscalerLoadBalancerNetworks listSrxFirewallNetworks updateNetwork	表中指令有三個response欄位，其他的沒有變 新增response欄位：causefordeploy, vpcid, tags(*)
createZone updateZone	表中指令有新增以下欄位，其他的沒有變 新增request欄位：localstorageenabled (選擇性) 新增response欄位：localstorageenabled
listZones	新增response欄位：localstorageenabled
rebootRouter changeServiceForRouter startRouter destroyRouter stopRouter	表中的指令有兩個新的response欄位，其他的沒有改變 新增response欄位：vpcid, nic(*)
updateAccount disableAccount listAccounts markDefaultZoneForAccount enableAccount	表中指令有三個response欄位，其他的沒有變 新增response欄位：vpcavailable, vpcimit, vpcototal
listRouters	新增request欄位：forvpc (選擇性), vpcid (選擇性) 新增response欄位：vpcid, nic(*)
listNetworkOfferings	新增request欄位：forvpc (選擇性) 新增response欄位：forvpc
listVolumes	新增request欄位：details (選擇性), tags (選擇性) 新的response欄位: tags(*)
addTrafficMonitor	新增request欄位：excludezones (非必須), includezones (非必須)
createNetwork	新增request欄位：vpcid (選擇性)

	新增response欄位：canusefordeploy, vpcid, tags(*)
listPublicIpsAddresses	新增request欄位：tags (選擇性), vpcid(選擇性) 新增response欄位：vpcid, tags(*)
listNetworks	新增request欄位：canusefordeploy (選擇性), forvpc (選擇性), tags (選擇性), vpcid (選擇性) 新增response欄位：canusefordeploy, vpcid, tags(*)
restartNetwork	新增response欄位：vpcid, tags(*)
enableStaticNat	新增request欄位：networkid(非必須)
createDiskOffering	新增request欄位：storagetype(非必須) 新增response欄位：storagetype
listDiskOfferings	新增response欄位：storagetype
updateDiskOffering	新增response欄位：storagetype
createFirewallRule	改變request欄位：ipaddressid (舊版 - 選擇性, 新版 - 需要) 新的response欄位：tags(*)
listVirtualMachines	新增request欄位：isoid (選擇性), tags (選擇性), templateid (選擇性) 新增response欄位：keypair, tags(*)
updateStorageNetworkIpsRange	新增response欄位：id, endip, gateway, netmask, networkid, podid, startup, vlan, zoneid

4.2.2. 4.0.0-incubating新增API指令

- › createCounter (增加counter)
- › deleteCounter (移除counter)
- › listCounters (條列counters)
- › createCondition (新增條件)
- › deleteCondition (移除condition)
- › listConditions (條列特定使用者的condition)
- › createTags. 為特定資源新增標籤, 例如：

```
command=createTags
&resourceIds=1,10,12
&resourceType=userVm
&tags[0].key=region
&tags[0].value=canada
&tags[1].key=city
&tags[1].value=Toronto
```

- › deleteTags. 移除選定資源標籤, 例如：

```
command=deleteTags
&resourceIds=1,12
&resourceType=Snapshot
&tags[0].key=city
```

- › listTags (描述選定資源標籤)
- › createVPC (新增VPC)
- › listVPCs (條列所有VPC)
- › deleteVPC (刪除一個VPC)
- › updateVPC (更新一個VPC)
- › restartVPC (重新啟動一個VPC)
- › createVPCOffering (新增一個VPC offering)
- › updateVPCOffering (更新一個VPC offering)
- › deleteVPCOffering (刪除一個VPC offering)
- › listVPCOfferings (條列所有的VPC offerings)
- › createPrivateGateway (新增私人關道)
- › listPrivateGateways (列出私人關道)
- › deletePrivateGateway (刪除私人關道)
- › createNetworkACL (新增ACL來規範網路(網路屬於VPC))
- › deleteNetworkACL (刪除網路ACL)
- › listNetworkACLs (列出所有網路ACL)
- › createStaticRoute (建立固定路徑)
- › deleteStaticRoute (刪除固定路徑)
- › listStaticRoutes (列出所有固定路徑)
- › createVpnCustomerGateway (新增 site to site VPN客戶關道)
- › createVpnGateway (建立 site to site VPN本地關道)
- › createVpnConnection (新增 site to site VPN連結)
- › deleteVpnCustomerGateway (刪除 site to site VPN訪客關道)
- › deleteVpnGateway (刪除 site to site VPN關道)
- › deleteVpnConnection (刪除 site to site VPN連結)
- › updateVpnCustomerGateway (更新 site to site VPN訪客關道)
- › resetVpnConnection (重設 site to site VPN連結)
- › listVpnCustomerGateways (列出 site to site VPN訪客關道)
- › listVpnGateways (列出 site 2 site VPN關道)
- › listVpnConnections (列出 site to site VPN連結關道)
- › enableCiscoNexusVSM (在CloudStack啟用 Nexus 1000v dvSwitch)
- › disableCiscoNexusVSM (在CloudStack停用 Nexus 1000v dvSwitch)
- › deleteCiscoNexusVSM (在CloudStack刪除Nexus 1000v dvSwitch)
- › listCiscoNexusVSMs (列出控制VLAN ID、封包VLAN ID及資料VLAN ID, 同時也列出Nexus 1000v dvSwitch的IP位址)

4.3. 3.0版本

4.3.1. 啟用8096網路連接埠

8096網路連接埠允許接受未經認證的API呼叫, 於3.0.1版本中預設為關閉狀態, 當您有類似以下需求時, 可以啟用8096網路連接埠(或是任何其他連接埠)：

1. 確保第一台Management Server已安裝並為運行狀態。
2. 設定global configuration參數, 指定integration.api.port的連接埠號。
3. 重新啟動 Management Server
4. 確認運行Management Server的主機其iptables的設定中, 允許此連接埠的通行。

4.3.2. 已停止的虛擬機器

在3.0.1版本中, 提供了停止虛擬機器但保留其狀態的功能。您可以指定要關閉的虛擬機器的名稱和ID, 如下：

CloudStack 提供了建立虛擬機器但仍為停止狀態的功能，你可以決定是否開啟虛擬機器為您佈署過程的一部分。一台虛擬機器可以有兩種情況(預設建立後啟用)，一是建立後啟用虛擬機器或是建立後不啟用虛擬機器。

deployVM指令提供了startVM的新參數，可以將您的虛擬機器於建立後保持關閉狀態。

可能的設定值為：

- › true - 於佈署虛擬機器之後開啟此虛擬機器。
- › false - 於佈署虛擬機器之後維持停止狀態。

預設值為true。

4.3.3. Change to Behavior of List Commands

There was a major change in how our List* API commands work in CloudStack 3.0 compared to 2.2.x. The rules below apply only for managed resources – those that belong to an account, domain, or project. They are irrelevant for the List* commands displaying unmanaged (system) resources, such as hosts, clusters, and external network resources.

When no parameters are passed in to the call, the caller sees only resources owned by the caller (even when the caller is the administrator). Previously, the administrator saw everyone else's resources by default.

When accountName and domainId are passed in:

- › The caller sees the resources dedicated to the account specified.
- › If the call is executed by a regular user, the user is authorized to specify only the user's own account and domainId.
- › If the caller is a domain administrator, CloudStack performs an authorization check to see whether the caller is permitted to view resources for the given account and domainId.

When projectId is passed in, only resources belonging to that project are listed.

When domainId is passed in, the call returns only resources belonging to the domain specified. To see the resources of subdomains, use the parameter isRecursive=true. Again, the regular user can see only resources owned by that user, the root administrator can list anything, and a domain administrator is authorized to see only resources of the administrator's own domain and subdomains.

To see all resources the caller is authorized to see, except for Project resources, use the parameter listAll=true.

To see all Project resources the caller is authorized to see, use the parameter projectId=-1.

There is one API command that doesn't fall under the rules above completely: the listTemplates command. This command has its own flags defining the list rules:

listTemplates Flag	敘述
featured	Returns templates that have been marked as featured and public.
self	Returns templates that have been registered or created by the calling user.
selfexecutable	Same as self, but only returns templates that are ready to be deployed with.
sharedexecutable	Ready templates that have been granted to the calling user by another user.
executable	Templates that are owned by the calling user, or public templates, that can be used to deploy a new VM.
community	Returns templates that have been marked as public but not featured.
all	Returns all templates (only usable by admins).

The CloudStack UI on a general view will display all resources that the logged-in user is authorized to see, except for project resources. To see the project resources, select the project view.

4.3.4. Removed API commands

- › createConfiguration (Adds configuration value)
- › configureSimulator (Configures simulator)

4.3.5. Added API commands in 3.0

4.3.5.1. Added in 3.0.2

- › changeServiceForSystemVm
Changes the service offering for a system VM (console proxy or secondary storage). The system VM must be in a "Stopped" state for this command to take effect.

4.3.5.2. Added in 3.0.1

- › changeServiceForSystemVm
Changes the service offering for a system VM (console proxy or secondary storage). The system VM must be in a "Stopped" state for this command to take effect.

4.3.5.3. Added in 3.0.0

assignVirtualMachine (Move a user VM to another user under same domain.)	restoreVirtualMachine (Restore a VM to original template or specific snapshot)	createLBStickinessPolicy (Creates a Load Balancer stickiness policy)
deleteLBStickinessPolicy (Deletes a LB stickiness policy.)	listLBStickinessPolicies (Lists LBStickiness policies.)	ldapConfig (Configure the LDAP context for this site.)
addSwift (Adds Swift)	listSwifts (List Swift.)	migrateVolume (Migrate volume)
updateStoragePool (Updates a storage pool.)	authorizeSecurityGroupEgress (Authorizes a particular egress rule for this security group)	revokeSecurityGroupEgress (Deletes a particular egress rule from this security group)
createNetworkOffering (Creates a network offering.)	deleteNetworkOffering (Deletes a network offering.)	createProject (Creates a project)
deleteProject (Deletes a project)	updateProject (Updates a project)	activateProject (Activates a project)
suspendProject (Suspends a project)	listProjects (Lists projects and provides detailed information for listed projects)	addAccountToProject (Adds account to a project)
deleteAccountFromProject (Deletes account from the project)	listProjectAccounts (Lists project's accounts)	listProjectInvitations (Lists an account's invitations to join projects)
updateProjectInvitation (Accepts or declines project invitation)	deleteProjectInvitation (Deletes a project invitation)	updateHypervisorCapabilities (Updates a hypervisor capabilities.)
listHypervisorCapabilities (Lists all hypervisor capabilities.)	createPhysicalNetwork (Creates a physical network)	deletePhysicalNetwork (Deletes a Physical Network.)
listPhysicalNetworks (Lists physical networks)	updatePhysicalNetwork (Updates a physical network)	listSupportedNetworkServices (Lists all network services provided by CloudStack or for the given Provider.)
addNetworkServiceProvider (Adds a network service provider to a physical network)	deleteNetworkServiceProvider (Deletes a Network Service Provider.)	listNetworkServiceProviders (Lists network service providers for a given physical network)

network		physical network
updateNetworkServiceProvider (Updates a network service provider of a physical network)	addTrafficType (Adds traffic type to a physical network)	deleteTrafficType (Deletes traffic type of a physical network)
listTrafficTypes (Lists traffic types of a given physical network)	updateTrafficType (Updates traffic type of a physical network)	listTrafficTypeImplementors (Lists implementors of implementor of a network traffic type or implementors of all network traffic types)
createStorageNetworkIpRange (Creates a Storage network IP range.)	deleteStorageNetworkIpRange (Deletes a storage network IP Range.)	listStorageNetworkIpRange (List a storage network IP range.)
updateStorageNetworkIpRange (Update a Storage network IP range, only allowed when no IPs in this range have been allocated.)	listUsageTypes (List Usage Types)	addF5LoadBalancer (Adds a F5 BigIP load balancer device)
configureF5LoadBalancer (configures a F5 load balancer device)	deleteF5LoadBalancer (delete a F5 load balancer device)	listF5LoadBalancers (lists F5 load balancer devices)
listF5LoadBalancerNetworks (lists network that are using a F5 load balancer device)	addSrxFirewall (Adds a SRX firewall device)	deleteSrxFirewall (delete a SRX firewall device)
listSrxFirewalls (lists SRX firewall devices in a physical network)	listSrxFirewallNetworks (lists network that are using SRX firewall device)	addNetscalerLoadBalancer (Adds a netscaler load balancer device)
deleteNetscalerLoadBalancer (delete a netscaler load balancer device)	configureNetscalerLoadBalancer (configures a netscaler load balancer device)	listNetscalerLoadBalancers (lists netscaler load balancer devices)
listNetscalerLoadBalancerNetworks (lists network that are using a netscaler load balancer device)	createVirtualRouterElement (Create a virtual router element.)	configureVirtualRouterElement (Configures a virtual router element.)
listVirtualRouterElements (Lists all available virtual router elements.)		

4.3.6. 增加的CloudStack錯誤碼

您可以在每種類型的例外中找到 CloudStack-specific 錯誤碼。以下列表為加到新class 的錯誤碼，class名稱為 CSEExceptionErrorCode

4250 : "com.cloud.utils.exception.CloudRuntimeException"	4255 : "com.cloud.utils.exception.ExceptionUtil"	4260 : "com.cloud.utils.exception.ExecutionException"
4265 : "com.cloud.utils.exception.HypervisorVersionChangedException"	4270 : "com.cloud.utils.exception.RuntimeCloudException"	4275 : "com.cloud.exception.CloudException"
4280 : "com.cloud.exception.AccountLimitException"	4285 : "com.cloud.exception.AgentUnavailableException"	4290 : "com.cloud.exception.CloudAuthenticationException"
4295 : "com.cloud.exception.CloudExecutionException"	4300 : "com.cloud.exception.ConcurrentOperationException"	4305 : "com.cloud.exception.ConflictingNetworkSettingsException"
4310 : "com.cloud.exception.DiscoveredWithErrorException"	4315 : "com.cloud.exception.HAStateException"	4320 : "com.cloud.exception.InsufficientAddressCapacityException"
4325 : "com.cloud.exception.InsufficientCapacityException"	4330 : "com.cloud.exception.InsufficientNetworkCapacityException"	4335 : "com.cloud.exception.InsufficientServerCapacityException"
4340 : "com.cloud.exception.InsufficientStorageCapacityException"	4345 : "com.cloud.exception.InternalErrorException"	4350 : "com.cloud.exception.InvalidParameterValueException"
4355 : "com.cloud.exception.ManagementServerException"	4360 : "com.cloud.exception.NetworkRuleConflictException"	4365 : "com.cloud.exception.PermissionDeniedException"
4370 : "com.cloud.exception.ResourceAllocationException"	4375 : "com.cloud.exception.ResourceInUseException"	4380 : "com.cloud.exception.ResourceUnavailableException"
4385 : "com.cloud.exception.StorageUnavailableException"	4390 : "com.cloud.exception.UnsupportedServiceException"	4395 : "com.cloud.exception.VirtualMachineMigrationException"
4400 : "com.cloud.exception.AccountLimitException"	4405 : "com.cloud.exception.AgentUnavailableException"	4410 : "com.cloud.exception.CloudAuthenticationException"
4415 : "com.cloud.exception.CloudException"	4420 : "com.cloud.exception.CloudExecutionException"	4425 : "com.cloud.exception.ConcurrentOperationException"
4430 : "com.cloud.exception.ConflictingNetworkSettingsException"	4435 : "com.cloud.exception.ConnectionException"	4440 : "com.cloud.exception.DiscoveredWithErrorException"
4445 : "com.cloud.exception.DiscoveryException"	4450 : "com.cloud.exception.HAStateException"	4455 : "com.cloud.exception.InsufficientAddressCapacityException"
4460 : "com.cloud.exception.InsufficientCapacityException"	4465 : "com.cloud.exception.InsufficientNetworkCapacityException"	4470 : "com.cloud.exception.InsufficientServerCapacityException"
4475 : "com.cloud.exception.InsufficientStorageCapacityException"	4480 : "com.cloud.exception.InsufficientVirtualNetworkCapacityException"	4485 : "com.cloud.exception.InternalErrorException"
4490 : "com.cloud.exception.InvalidParameterValueException"	4495 : "com.cloud.exception.ManagementServerException"	4500 : "com.cloud.exception.NetworkRuleConflictException"
4505 : "com.cloud.exception.PermissionDeniedException"	4510 : "com.cloud.exception.ResourceAllocationException"	4515 : "com.cloud.exception.ResourceInUseException"
4520 : "com.cloud.exception.ResourceUnavailableException"	4525 : "com.cloud.exception.StorageUnavailableException"	4530 : "com.cloud.exception.UnsupportedServiceException"
4535 : "com.cloud.exception.VirtualMachineMigrationException"	9999 : "org.apache.cloudstack.api.ServerApiException"	

章 5. 對CloudStack API 發出請求

5.1. Making API Requests

5.2. Signing API Requests

5.2.1. How to sign an API call with Python

5.3. Enabling API Call Expiration

5.4. Limiting the Rate of API Requests

5.4.1. Configuring the API Request Rate

5.4.2. Limitations on API Throttling

5.5. Responses

5.5.1. Response Formats: XML and JSON

5.5.2. Maximum Result Pages Returned

5.5.3. Error Handling

5.6. Asynchronous Commands

5.6.1. 工作狀態

5.6.2. Example

5.1. Making API Requests

All CloudStack API requests are submitted in the form of a HTTP GET/POST with an associated command and any parameters. A request is composed of the following whether in HTTP or HTTPS:

- CloudStack API URL: This is the web services API entry point (for example, <http://www.cloud.com:8080/client/api>)

- ▶ **Command:** The web services command you wish to execute, such as start a virtual machine or create a disk volume
- ▶ **Parameters:** Any additional required or optional parameters for the command

As sample API GET request looks like the following:

```
http://localhost:8080/client/api?
command=deployVirtualMachine&serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId=4&api
Key=miVr6X7u6bN_sdah0BpjNejPgEsT35eXq-
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001h
U%3D
```

Or in a more readable format:

```
1. http://localhost:8080/client/api
2. ?command=deployVirtualMachine
3. &serviceOfferingId=1
4. &diskOfferingId=1
5. &templateId=2
6. &zoneId=4
7.
&apiKey=miVr6X7u6bN_sdah0BpjNejPgEsT35eXqjB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikw
FQ
8. &signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU%3D
```

The first line is the CloudStack API URL. This is the Cloud instance you wish to interact with.

The second line refers to the command you wish to execute. In our example, we are attempting to deploy a fresh new virtual machine. It is preceded by a (?) to separate itself from the CloudStack API URL.

Lines 3-6 are the parameters for this given command. To see the command and its request parameters, please refer to the appropriate section in the CloudStack API documentation. Each parameter field-value pair (field=value) is preceded by an ampersand character (&).

Line 7 is the user API Key that uniquely identifies the account. See [Signing API Requests](#) on page 7.

Line 8 is the signature hash created to authenticate the user account executing the API command. See [Signing API Requests](#) on page 7.

5.2. Signing API Requests

Whether you access the CloudStack API with HTTP or HTTPS, it must still be signed so that CloudStack can verify the caller has been authenticated and authorized to execute the command. Make sure that you have both the API Key and Secret Key provided by the CloudStack administrator for your account before proceeding with the signing process.

To show how to sign a request, we will re-use the previous example.

```
http://http://localhost:8080/client/api?
command=deployVirtualMachine&serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId=4&api
Key=miVr6X7u6bN_sdah0BpjNejPgEsT35eXq-
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001h
U%3D
```

Breaking this down, we have several distinct parts to this URL.

- ▶ **Base URL:** This is the base URL to the CloudStack Management Server.

```
http://localhost:8080
```

- ▶ **API Path:** This is the path to the API Servlet that processes the incoming requests.

```
/client/api?
```

- ▶ **Command String:** This part of the query string comprises of the command, its parameters, and the API Key that identifies the account.

注意

As with all query string parameters of field-value pairs, the "field" component is case insensitive while all "value" values are case sensitive.

```
command=deployVirtualMachine&serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId=4&
apiKey=miVr6X7u6bN_sdah0BpjNejPgEsT35eXq-
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ
```

- ▶ **Signature:** This is the signature of the command string that is generated using a combination of the user's Secret Key and the HMAC SHA-1 hashing algorithm.

```
&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU%3D
```

Every API request has the format Base URL+API Path+Command String+Signature.

To generate the signature.

1. For each field-value pair (as separated by a '&') in the Command String, URL encode each value so that it can be safely sent via HTTP GET.

注意

Make sure all spaces are encoded as "%20" rather than "+".

2. Lower case the entire Command String and sort it alphabetically via the field for each field-value pair. The result of this step would look like the following.

```
apikey=miVr6X7u6bN_sdah0BpjNejPgEsT35eXq-
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ&command=deployVirtualMachine&dis
kOfferingId=1&serviceOfferingId=1&templateId=2&zoneId=4
```

3. Take the sorted Command String and run it through the HMAC SHA-1 hashing algorithm (most programming languages offer a utility method to do this) with the user's Secret Key. Base64 encode the resulting byte array in UTF-8 so that it can be safely transmitted via HTTP. The final string produced after Base64 encoding should be "Lxx1DM40AjcXU%2FcaiK8RAP001hU%3D".

By reconstructing the final URL in the format Base URL+API Path+Command String+Signature, the final URL should look like:

```
http://localhost:8080/client/api?
command=deployVirtualMachine&serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId
=4&apiKey=miVr6X7u6bN_sdah0BpjNejPgEsT35eXq-
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ&signature=Lxx1DM40AjcXU%2FcaiK8R
AP001hU%3D
```

5.2.1. How to sign an API call with Python

To illustrate the procedure used to sign API calls we present a step by step interactive session using Python.

First import the required modules:

```

$python
Python 2.7.3 (default, Nov 17 2012, 19:54:34)
[GCC 4.2.1 Compatible Apple Clang 4.1 ((tags/Apple/clang-421.11.66))] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import urllib2
>>> import urllib
>>> import hashlib
>>> import hmac
>>> import base64

```

Define the endpoint of the Cloud, the command that you want to execute and the keys of the user.

```

>>> baseurl='http://localhost:8080/client/api?'
>>> request={}
>>> request['command']='listUsers'
>>> request['response']='json'
>>> request['apikey']='plgWJfZK4gyS3m0MTvmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
kAYMmAlLqzzq1E1ZLYq_u38zCm0bewzGUDP66mg'
>>>
secretkey='VdaACyB0LV9eNjTetIOElcVQkvJck_J_QljX_FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3a
kAK8Q'

```

Build the request string:

```

>>> request_str='&'.join(['=' .join([k,urllib.quote_plus(request[k])]) for k in
request.keys()])
>>> request_str
'apikey=plgWJfZK4gyS3m0MTvmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
kAYMmAlLqzzq1E1ZLYq_u38zCm0bewzGUDP66mg&command=listUsers&response=json'

```

Compute the signature with hmac, do a 64 bit encoding and a url encoding:

```

>>>
sig_str='&'.join(['=' .join([k.lower(),urllib.quote_plus(request[k].lower().replace('+','%20'))])for k in sorted(request.iterkeys())])
>>> sig_str
'apikey=plgWJfZK4gyS3m0MTvmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
kaymmailqzzq1e1zlyq_u38zcm0bewzgudp66mg&command=listusers&response=json'
>>> sig=hmac.new(secretkey,sig_str,hashlib.sha1)
>>> sig
<hmac.HMAC instance at 0x10d91d680>
>>> sig=hmac.new(secretkey,sig_str,hashlib.sha1).digest()
>>> sig
'M:J\x0e\xaf\xfb\x8f\x2y\xf1p\x91\x1e\x89\x8a\xa1\x05\xc4A\xdb'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest())
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds=\n'
>>> sig=base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest()).strip()
>>> sig
'TTpdDq/7j/J58XCRHomKoQXEQds='
>>>
sig=urllib.quote_plus(base64.encodestring(hmac.new(secretkey,sig_str,hashlib.sha1).digest())
).strip()

```

Finally, build the entire string and do an http GET:

```

>>> req=baseurl+request_str+'&signature='+sig
>>> req
'http://localhost:8080/client/api?apikey=plgWJfZK4gyS3m0MTvmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
kAYMmAlLqzzq1E1ZLYq_u38zCm0bewzGUDP66mg&command=listUsers&response=json&signature=TTpdDq/7j/J58XCRHomKoQXEQds=3D'
>>> res=urllib2.urlopen(req)
>>> res.read()
{ "listUsersResponse" : { "count":3, "user" : [ { "id": "7ed6d5da-93b2-4545-a502-
23d20b40ef2a", "username": "admin", "firstname": "admin", "lastname": "cLoud", "created": "2012-07-
05T12:18:27-0700", "state": "enabled", "account": "admin", "accounttype": "1", "domainid": "8a111e58-
e155-4482-93ce-84efff3c7c77", "domain": "ROOT", "apikey": "plgWJfZK4gyS3m0MTvmjUVg-X-
jlWlnfaUJ9GAbBbf9EdM-
kAYMmAlLqzzq1E1ZLYq_u38zCm0bewzGUDP66mg", "secretkey": "VdaACyB0LV9eNjTetIOElcVQkvJck_J_QljX_
FcHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akAK8Q", "accountid": "7548ac03-af1d-4c1c-9064-
2f3e2c0eda0d"}, { "id": "1fea6418-5576-4989-a21e-
4790787bbee3", "username": "runseb", "firstname": "foobar", "lastname": "goa", "email": "joe@smith.
com", "created": "2013-04-10T16:52:06-
0700", "state": "enabled", "account": "admin", "accounttype": "1", "domainid": "8a111e58-e155-4482-
93ce-
84efff3c7c77", "domain": "ROOT", "apikey": "Xhsb3MewjJQaXXMsZrLVLQI9_NPy_UcbDj1QXikkVbDC9MDSpWw
dt21DUY1H7JBEYTDDLY3yuchCeW778GkBA", "secretkey": "gIsgmi8C5YwMHjX5o51pSe0kqs6JnKriw0jJBLce
Y5bgnfZkjl4aM6ctJX-1iddQIHJLbLJK9MRzSKk6xZ_w", "accountid": "7548ac03-af1d-4c1c-9064-
2f3e2c0eda0d"}, { "id": "52f65396-183c-4473-883f-
a37e7bb93967", "username": "toto", "firstname": "john", "lastname": "smith", "email": "john@smith.c
om", "created": "2013-04-23T04:27:22-
0700", "state": "enabled", "account": "admin", "accounttype": "1", "domainid": "8a111e58-e155-4482-
93ce-
84efff3c7c77", "domain": "ROOT", "apikey": "ThA6fFWS_OmvU8od201omXFC8yKNL_Hc5ZCS77LFCJsRzSx48J
yZucbuU16XYbEg-ZyXm1_wuEpECzK-
wKknow", "secretkey": "05ywpqJorAseBKR_5jEvrt6HfWL1j_1e4Z_iCr80KCYcsPI0dVcfzjJQ8YqK0a5EzSpOrR
j0FiLsg0hQrYnDA", "accountid": "7548ac03-af1d-4c1c-9064-2f3e2c0eda0d" } ] } }

```

5.3. Enabling API Call Expiration

You can set an expiry timestamp on API calls to prevent replay attacks over non-secure channels, such as HTTP. The server tracks the expiry timestamp you have specified and rejects all the subsequent API requests that come in after this validity period.

To enable this feature, add the following parameters to the API request:

- signatureVersion=3: If the signatureVersion parameter is missing or is not equal to 3, the expires parameter is ignored in the API request.
- expires=YYYY-MM-DDThh:mm:ssZ: Specifies the date and time at which the signature included in the request is expired. The timestamp is expressed in the YYYY-MM-DDThh:mm:ssZ format, as specified in the ISO 8601 standard.

例如：

```
expires=2011-10-10T12:00:00+0530
```

A sample API request with expiration is given below:

```

http://<IPAddress>:8080/client/api?command=listZones&signatureVersion=3&expires=2011-10-
10T12:00:00+0530&apiKey=m1Vr6X7u6bn_sdah0BpjNepPgEsT35eXq-
jB8CG20Y13yaxXcpgyuaIRmFI_EJTjVwZ0nUkkJbPmy3y2bc1kWFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001h
U%3D

```

5.4. Limiting the Rate of API Requests

You can limit the rate at which API requests can be placed for each account. This is useful to avoid malicious attacks on the Management Server, prevent performance degradation, and provide fairness to all accounts.

If the number of API calls exceeds the threshold, an error message is returned for any additional API calls. The caller will have to retry these API calls at another time.

5.4.1. Configuring the API Request Rate

To control the API request rate, use the following global configuration settings:

- › `api.throttling.enabled` - Enable/Disable API throttling. By default, this setting is false, so API throttling is not enabled.
- › `api.throttling.interval` (in seconds) - Time interval during which the number of API requests is to be counted. When the interval has passed, the API count is reset to 0.
- › `api.throttling.max` - Maximum number of APIs that can be placed within the `api.throttling.interval` period.
- › `api.throttling.cachesize` - Cache size for storing API counters. Use a value higher than the total number of accounts managed by the cloud. One cache entry is needed for each account, to store the running API total for that account.

5.4.2. Limitations on API Throttling

The following limitations exist in the current implementation of this feature.



注意

Even with these limitations, CloudStack is still able to effectively use API throttling to avoid malicious attacks causing denial of service.

- › In a deployment with multiple Management Servers, the cache is not synchronized across them. In this case, CloudStack might not be able to ensure that only the exact desired number of API requests are allowed. In the worst case, the number of API calls that might be allowed is (number of Management Servers) * (api.throttling.max).
- › The API commands `resetApiLimit` and `getApiLimit` are limited to the Management Server where the API is invoked.

5.5. Responses

5.5.1. Response Formats: XML and JSON

CloudStack supports two formats as the response to an API call. The default response is XML. If you would like the response to be in JSON, add `&response=json` to the Command String.

The two response formats differ in how they handle blank fields. In JSON, if there is no value for a response field, it will not appear in the response. If all the fields were empty, there might be no response at all. In XML, even if there is no value to be returned, an empty field will be returned as a placeholder XML element.

Sample XML Response:

```
<listipaddressesresponse>
  <allocatedipaddress>
    <ipaddress>192.168.10.141</ipaddress>
    <allocated>2009-09-18T13:16:10-0700</allocated>
    <zoneid>4</zoneid>
    <zonename>WC</zonename>
    <issourcenat>true</issourcenat>
  </allocatedipaddress>
</listipaddressesresponse>
```

Sample JSON Response:

```
{ "listipaddressesresponse" :
  { "allocatedipaddress" :
    [
      {
        "ipaddress" : "192.168.10.141",
        "allocated" : "2009-09-18T13:16:10-0700",
        "zoneid" : "4",
        "zonename" : "WC",
        "issourcenat" : "true"
      }
    ]
  }
}
```

5.5.2. Maximum Result Pages Returned

For each cloud, there is a default upper limit on the number of results that any API command will return in a single page. This is to help prevent overloading the cloud servers and prevent DOS attacks. For example, if the page size limit is 500 and a command returns 10,000 results, the command will return 20 pages.

The default page size limit can be different for each cloud. It is set in the global configuration parameter `default.page.size`. If your cloud has many users with lots of VMs, you might need to increase the value of this parameter. At the same time, be careful not to set it so high that your site can be taken down by an enormous return from an API call. For more information about how to set global configuration parameters, see "Describe Your Deployment" in the Installation Guide.

To decrease the page size limit for an individual API command, override the global setting with the `page` and `pagesize` parameters, which are available in any `list*` command (`listCapabilities`, `listDiskOfferings`, etc.).

- › Both parameters must be specified together.
- › The value of the `pagesize` parameter must be smaller than the value of `default.page.size`. That is, you can not increase the number of possible items in a result page, only decrease it.

For syntax information on the `list*` commands, see the API Reference.

5.5.3. Error Handling

If an error occurs while processing an API request, the appropriate response in the format specified is returned. Each error response consists of an error code and an error text describing what possibly can go wrong. For an example error response, see page 12.

An HTTP error code of 401 is always returned if API request was rejected due to bad signatures, missing API Keys, or the user simply did not have the permissions to execute the command.

5.6. Asynchronous Commands

Asynchronous commands were introduced in CloudStack 2.x. Commands are designated as asynchronous when they can potentially take a long period of time to complete such as creating a snapshot or disk volume. They differ from synchronous commands by the following:

- › They are identified in the API Reference by an (A).
- › They will immediately return a job ID to refer to the job that will be responsible in processing the command.
- › If executed as a "create" resource command, it will return the resource ID as well as the job ID.

You can periodically check the status of the job by making a simple API call to the command, `queryAsyncJobResult` and passing in the job ID.

5.6.1. 工作狀態

使用非同步指令的跡象為工作ID, 此ID會在指令執行時立即回覆。當您取得ID, 您可以隨時檢查工作狀態, 使用 `queryAsyncJobResult` 指令, 此指令會回復三種可能的工作狀態正數:

- ▶ 0: 工作正在執行, 繼續週期性的檢查
- ▶ 1: 工作完成, 會回復任何成功的回應數值, 與一開始執行的指令有關
- ▶ 2: 工作執行失敗, 請檢查"jobresultcode"標籤檢查失敗原因的程式碼, 及"jobresult"檢查失敗原因

5.6.2. Example

The following shows an example of using an asynchronous command. Assume the API command:

```
command=deployVirtualMachine&zoneId=1&serviceOfferingId=1&diskOfferingId=1&templateId=1
```

CloudStack will immediately return a job ID and any other additional data.

```
<deployvirtualmachineresponse>
  <jobid>1</jobid>
  <id>100</id>
</deployvirtualmachineresponse>
```

Using the job ID, you can periodically poll for the results by using the `queryAsyncJobResult` command.

```
command=queryAsyncJobResult&jobId=1
```

Three possible results could come from this query.

Job is still pending:

```
<queryasyncjobresult>
  <jobid>1</jobid>
  <jobstatus>0</jobstatus>
  <jobprocstatus>1</jobprocstatus>
</queryasyncjobresult>
```

Job has succeeded:

```
<queryasyncjobresultresponse cloud-stack-version="3.0.1.6">
  <jobid>1</jobid>
  <jobstatus>1</jobstatus>
  <jobprocstatus>0</jobprocstatus>
  <jobresultcode>0</jobresultcode>
  <jobresulttype>object</jobresulttype>
  <jobresult>
    <virtualmachine>
      <id>450</id>
      <name>i-2-450-VM</name>
      <displayname>i-2-450-VM</displayname>
      <account>admin</account>
      <domainid>1</domainid>
      <domain>ROOT</domain>
      <created>2011-03-10T18:20:25-0800</created>
      <state>Running</state>
      <haenable>false</haenable>
      <zoneid>1</zoneid>
      <zonename>San Jose 1</zonename>
      <hostid>2</hostid>
      <hostname>905-13.sjc.lab.vmops.com</hostname>
      <templateid>1</templateid>
      <templatename>CentOS 5.3 64bit LAMP</templatename>
      <templatedisplaytext>CentOS 5.3 64bit LAMP</templatedisplaytext>
      <passwordenabled>false</passwordenabled>
      <serviceofferingid>1</serviceofferingid>
      <serviceofferingname>Small Instance</serviceofferingname>
      <cpunumber>1</cpunumber>
      <cpuspeed>500</cpuspeed>
      <memory>512</memory>
      <guestosid>12</guestosid>
      <rootdeviceid>0</rootdeviceid>
      <rootdevicetype>NetworkFilesystem</rootdevicetype>
      <nic>
        <id>561</id>
        <networkid>205</networkid>
        <netmask>255.255.255.0</netmask>
        <gateway>10.1.1.1</gateway>
        <ipaddress>10.1.1.225</ipaddress>
        <isolationuri>vlan://295</isolationuri>
        <broadcasturi>vlan://295</broadcasturi>
        <traffictype>Guest</traffictype>
        <type>Virtual</type>
        <isdefault>true</isdefault>
      </nic>
      <hypervisor>XenServer</hypervisor>
    </virtualmachine>
  </jobresult>
</queryasyncjobresultresponse>
```

Job has failed:

```
<queryasyncjobresult>
  <jobid>1</jobid>
  <jobstatus>2</jobstatus>
  <jobprocstatus>0</jobprocstatus>
  <jobresultcode>551</jobresultcode>
  <jobresulttype>text</jobresulttype>
  <jobresult>Unable to deploy virtual machine id = 100 due to not enough
capacity</jobresult>
</queryasyncjobresult>
```

章 6. 使用使用資料

6.1. Usage Record Format

- 6.1.1. Virtual Machine Usage Record Format
- 6.1.2. Network Usage Record Format
- 6.1.3. IP位址使用紀錄的格式
- 6.1.4. Disk Volume Usage Record Format
- 6.1.5. 模組、ISO及快取物件的使用紀錄格式
- 6.1.6. Load Balancer Policy 或 Port Forwarding Rule 的使用紀錄格式
- 6.1.7. Network Offering Usage Record Format
- 6.1.8. VPN使用者使用紀錄格式

6.2. Usage Types

6.3. Example response from listUsageRecords

6.4. Dates in the Usage Record

6.5. 廣域設定限制

使用伺服器提供彙整使用紀錄，讓您可以在CloudStack上建立廣告整合。使用伺服器會從事件紀錄擷取資料，然後創立使用紀錄總結，您可以用listUsageRecords API call來存取

使用紀錄顯示資源的數量，像是VM執行時間或guest instances消耗的模組儲存空間。對於裸金 instances 這個特例，沒有消耗模組儲存資源，但會紀錄帶使用率，並顯示在使用伺服器的使用結果

使用伺服器至少每天執行一次，也可以透過設定，每天執行多次。其行為由 CloudStack 管理者指南中描述的系統設定控制

6.1. Usage Record Format

6.1.1. Virtual Machine Usage Record Format

For running and allocated virtual machine usage, the following fields exist in a usage record:

- › account：帳戶名稱
- › accountid：帳戶的ID
- › domainid：帳戶所在的網域ID
- › zoneid：使用的區域
- › description：使用紀錄追蹤的字串描述
- › usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for VM running time)
- › usagetype：代表使用形式的數字(詳見Usage Types)
- › rawusage：代表實際使用率的數字，單位為小時
- › virtualMachineid – The ID of the virtual machine
- › name – The name of the virtual machine
- › offeringid – The ID of the service offering
- › templateid – The ID of the template or the ID of the parent template. The parent template value is present when the current template was created from a volume.
- › usageid – Virtual machine
- › type – Hypervisor
- › startdate, enddate：使用總計的時間，詳見Dates in the Usage Record

6.1.2. Network Usage Record Format

For network usage (bytes sent/received), the following fields exist in a usage record.

- › account：帳戶名稱
- › accountid：帳戶的ID
- › domainid：帳戶所在的網域ID
- › zoneid：使用的區域
- › description：使用紀錄追蹤的字串描述
- › usagetype：代表使用形式的數字(詳見Usage Types)
- › rawusage：代表實際使用率的數字，單位為小時
- › usageid – Device ID (virtual router ID or external device ID)
- › type – Device type (domain router, external load balancer, etc.)
- › startdate, enddate：使用總計的時間，詳見Dates in the Usage Record

6.1.3. IP位址使用紀錄的格式

以下欄位會出現在使用紀錄中

- › account：帳戶名稱
- › accountid：帳戶的ID
- › domainid：帳戶所在的區域ID
- › zoneid：使用的地區
- › description：描述使用紀錄在追蹤甚麼的字串
- › usage：代表使用率的字串，包括單位
- › usagetype：代表使用形式的數字(詳見 Usage Types)
- › rawusage：帶表實際使用率的數字，單位為小時
- › usageid：IP位址的ID
- › startdate, enddate：使用總計的時間，詳見Dates in the Usage Record
- › issourcenat：source NAT是否啟用
- › iselastic：如果IP位址是彈性的，顯示True

6.1.4. Disk Volume Usage Record Format

For disk volumes, the following fields exist in a usage record.

- › account：帳戶名稱
- › accountid：帳戶的ID
- › domainid：帳戶所在的網域ID
- › zoneid：使用的區域
- › description：使用紀錄追蹤的字串描述
- › usage：代表使用率的字串，包括單位(e.g. 'Hrs' 代表小時)
- › usagetype：代表使用形式的數字(詳見Usage Types)
- › rawusage：代表實際使用率的數字，單位為小時
- › usageid – The volume ID
- › offeringid：硬碟服務的ID
- › type – Hypervisor
- › templateid – ROOT template ID
- › size – The amount of storage allocated
- › startdate, enddate：使用總計的時間，詳見Dates in the Usage Record

6.1.5. 模組、ISO及快取物件的使用紀錄格式

- › account：帳戶名稱
- › accountid：帳戶的ID
- › domainid：帳戶所在的網域ID
- › zoneid：使用的區域
- › description：使用紀錄追蹤的字串描述

- › usage : 代表使用率的字串, 包括單位(e.g. 'Hrs' 代表小時)
- › usagetype : 代表使用形式的數字(詳見Usage Types)
- › rawusage : 代表實際使用率的數字, 單位為小時
- › usageid : 模組、ISO或快取物件的ID
- › offeringid : 硬碟服務的ID
- › templateid : 僅模組(usage type 7)包含。來源模組的ID
- › size : 模組、ISO或快取物件的大小
- › startdate, enddate : 使用總計的時間, 詳見Dates in the Usage Record

6.1.6. Load Balancer Policy 或 Port Forwarding Rule 的使用季錄格式

- › account : 帳戶名稱
- › accountid : 帳戶的ID
- › domainid : 帳戶所在的區域ID
- › zoneid : 使用的地區
- › description : 描述使用紀錄在追蹤甚麼的字串
- › usage : 代表使用率的字串, 包括單位(e.g. 'Hrs' for hours)
- › usagetype : 代表使用形式的數字(詳見)
- › rawusage : 帶表實際使用率的數字, 單位為小時
- › usageid : load balancer policy 或 port forwarding rule的ID
- › usagetype : 代表使用形式的數字(詳見)
- › startdate, enddate : 使用總計的時間, 詳見Dates in the Usage Record

6.1.7. Network Offering Usage Record Format

- › account : 帳戶名稱
- › accountid : 帳戶的ID
- › domainid : 帳戶所在的網域ID
- › zoneid : 使用的區域
- › description : 使用紀錄追蹤的字串描述
- › usage : 代表使用率的字串, 包括單位(e.g. 'Hrs' 代表小時)
- › usagetype : 代表使用形式的數字(詳見Usage Types)
- › rawusage : 代表實際使用率的數字, 單位為小時
- › usageid – ID of the network offering
- › usagetype : 代表使用形式的數字(詳見Usage Types)
- › offeringid – Network offering ID
- › virtualMachineid – The ID of the virtual machine
- › virtualMachineid – The ID of the virtual machine
- › startdate, enddate : 使用總計的時間, 詳見Dates in the Usage Record

6.1.8. VPN使用者使用紀錄格式

- › account : 帳戶名稱
- › accountid : 帳戶的ID
- › domainid : 帳戶所在的網域ID
- › zoneid : 使用的區域
- › description : 使用紀錄追蹤的字串描述
- › usage : 代表使用率的字串, 包括單位(e.g. 'Hrs' 代表小時)
- › usagetype : 代表使用形式的數字(詳見Usage Types)
- › rawusage : 代表實際使用率的數字, 單位為小時
- › usageid : VPN使用者的ID
- › usagetype : 代表使用形式的數字(詳見Usage Types)
- › startdate, enddate : 使用總計的時間, 詳見Dates in the Usage Record

6.2. Usage Types

The following table shows all usage types.

Type ID	Type Name	敘述
1	RUNNING_VM	Tracks the total running time of a VM per usage record period. If the VM is upgraded during the usage period, you will get a separate Usage Record for the new upgraded VM.
2	ALLOCATED_VM	Tracks the total time the VM has been created to the time when it has been destroyed. This usage type is also useful in determining usage for specific templates such as Windows-based templates.
3	IP_ADDRESS	Tracks the public IP address owned by the account.
4	NETWORK_BYTES_SENT	Tracks the total number of bytes sent by all the VMs for an account. Cloud.com does not currently track network traffic per VM.
5	NETWORK_BYTES_RECEIVED	Tracks the total number of bytes received by all the VMs for an account. Cloud.com does not currently track network traffic per VM.
6	VOLUME	Tracks the total time a disk volume has been created to the time when it has been destroyed.
7	TEMPLATE	Tracks the total time a template (either created from a snapshot or uploaded to the cloud) has been created to the time it has been destroyed. The size of the template is also returned.
8	ISO	Tracks the total time an ISO has been uploaded to the time it has been removed from the cloud. The size of the ISO is also returned.
9	SNAPSHOT	Tracks the total time from when a snapshot has been created to the time it have been destroyed.
11	LOAD_BALANCER_POLICY	Tracks the total time a load balancer policy has been created to the time it has been removed. Cloud.com does not track whether a VM has been assigned to a policy.
12	PORT_FORWARDING_RULE	Tracks the time from when a port forwarding rule was created until the time it was removed.
13	NETWORK_OFFERING	The time from when a network offering was assigned to a VM until it is removed.
14	VPN_USERS	The time from when a VPN user is created until it is removed.

6.3. Example response from listUsageRecords

All CloudStack API requests are submitted in the form of a HTTP GET/POST with an associated command and any parameters. A request is composed of the following whether in HTTP or HTTPS:

parameters requests composed of the following fields in the response:

```
<listusagerecordsresponse>
  <count>1816</count>
  <usage-record>
    <account>user5</account>
    <accountid>10004</accountid>
    <domainid>1</domainid>
    <zoneid>1</zoneid>
    <description>i-3-4-WC running time (ServiceOffering: 1) (Template:
3)</description>
    <usage>2.95288 Hrs</usage>
    <usagetype>1</usagetype>
    <rawusage>2.95288</rawusage>
    <virtualmachineid>4</virtualmachineid>
    <name>i-3-4-WC</name>
    <offeringid>1</offeringid>
    <templateid>3</templateid>
    <usageid>245554</usageid>
    <type>XenServer</type>
    <startdate>2009-09-15T00:00:00-0700</startdate>
    <enddate>2009-09-18T16:14:26-0700</enddate>
  </usage-record>
  ... (1,815 more usage records)
</listusagerecordsresponse>
```

6.4. Dates in the Usage Record

Usage records include a start date and an end date. These dates define the period of time for which the raw usage number was calculated. If daily aggregation is used, the start date is midnight on the day in question and the end date is 23:59:59 on the day in question (with one exception; see below). A virtual machine could have been deployed at noon on that day, stopped at 6pm on that day, then started up again at 11pm. When usage is calculated on that day, there will be 7 hours of running VM usage (usage type 1) and 12 hours of allocated VM usage (usage type 2). If the same virtual machine runs for the entire next day, there will 24 hours of both running VM usage (type 1) and allocated VM usage (type 2).

Note: The start date is not the time a virtual machine was started, and the end date is not the time when a virtual machine was stopped. The start and end dates give the time range within which usage was calculated.

For network usage, the start date and end date again define the range in which the number of bytes transferred was calculated. If a user downloads 10 MB and uploads 1 MB in one day, there will be two records, one showing the 10 megabytes received and one showing the 1 megabyte sent.

There is one case where the start date and end date do not correspond to midnight and 11:59:59pm when daily aggregation is used. This occurs only for network usage records. When the usage server has more than one day's worth of unprocessed data, the old data will be included in the aggregation period. The start date in the usage record will show the date and time of the earliest event. For other types of usage, such as IP addresses and VMs, the old unprocessed data is not included in daily aggregation.

6.5. 廣域設定限制

一個區域內，訪客虛擬網路預設上有24位元的CIDR，此限制訪客虛擬機器僅能執行254個instance。可依需要調整，但是必須在任何instance加到區域前調整。例如，10.1.1.0/22會提供~1000個位址

以下表格列出廣域組態的限制：

欄位名稱	定義
max.account.public.ips	可以被一個帳戶擁有的公開IP數量
max.account.snapshots	一個帳戶中的最多快照物件數量
max.account.templates	一個帳戶中的最多模板數量
max.account.user.vms	一個帳戶中的最多虛擬機器 instance數量
max.account.volumes	一個帳戶中的最多硬碟容量數量
max.template.iso.size	下載的模組或ISO的最大大小，單位為GB
max.volume.size.gb	容量最大的大小，單位為GB
network.throttling.rate	預設的資料傳輸速率，單位為MBit/s (支援XenServer)
snapshot.max.hourly	容量每小時重複快取的最大數量，如果快達限制，較早的快取物件會被刪除。此限制不適用手動快取。如果設為0，每小時重複快取將無法排程
snapshot.max.daily	容量每天重複快取的最大數量，如果快達限制，較早的快取物件會被刪除。此限制不適用手動快取。如果設為0，每天重複快取將無法排程
snapshot.max.weekly	容量每周重複快取的最大數量，如果快達限制，較早的快取物件會被刪除。此限制不適用手動快取。如果設為0，每周重複快取將無法排程
snapshot.max.monthly	容量每月重複快取的最大數量，如果快達限制，較早的快取物件會被刪除。此限制不適用手動快取。如果設為0，每月重複快取將無法排程

使用CloudStack使用者介面的廣域組態視窗來修改廣域組態參數欄位。詳見Setting Global Configuration Parameters

章 7. 準備及建立CloudStack文件

7.1. 安裝

7.2. 建立CloudStack文件

7.3. 撰寫CloudStack文件

7.4. 翻譯CloudStack文件

7.4.1. 翻譯CloudStack文件

這個章節敘述如何安裝publican、如何寫新的文件、建立指南及如何使用transifex建立翻譯版本

7.1. 安裝

CloudStack文件使用Publican建立，這個章節敘述如何安裝在您的機器上



注意

CloudStack文件原始碼位於
Publican文件也非常useful

在RHEL和RHEL衍生的東西，用下列指令安裝：

```
yum install publican publican-doc
```

在 Ubuntu, 用以下指令安裝:

```
apt-get install publican publican-doc
```

其他產品列在上面, 對於最新版的OSX, 您需要從 [source](#) 安裝, 並轉成您自己的設定

當publican安裝好了, 您需要設定CloudStack brand, 定義在docs/publican-CloudStack

進入以下指令:

```
sudo cp -R publican-cloudstack /usr/share/publican/Common_Content/cloudstack
```

如果失敗或您之後看見錯誤, 請見publican [documentation](#)

當您publican和CloudStack brand檔案都建好了, 您可以建立任何文件指南

7.2. 建立CloudStack文件

要建立特別的指南, 請到docs中文件的source tree並確定您要建立的指南

Currently there are four guides plus the release notes, all defined in publican configuration files:

```
publican-adminguide.cfg
publican-devguide.cfg
publican-installation.cfg
publican-plugin-niciranvp.cfg
publican-release-notes.cfg
```

要建立Developer指南, 執行以下:

```
publican build --config=publican-devguide.cfg --formats=pdf --langs=en-US
```

有一個PDF檔案會以 tmp/en-US/pdf建立, 您或許想建立其他格式, 例如html, 那麼您只要改變格式的直即可

7.3. 撰寫CloudStack文件

CloudStack文件格式為DocBook xml, 每份指南以參考DocBook *book*的publican configuration file 定義

這些書以xml files in docs/en-US定義, 因此, 如果我們翻閱Developers指南, 它的設定文件應包含:

```
xml_lang: en-US
type: Book
docname: Developers_Guide
brand: cloudstack
chunk_first: 1
chunk_section_depth: 1
```

The *docname* key gives you the basename of the DocBook file located in the en-US directory that contains the description of the book.

Looking closely at Developers_Guide.xml we see that it contains *book* tags and several references to other xml files. These are the chapters of the book, currently they are:

```
<xi:include href="concepts.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="building-with-maven.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="developer-introduction.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="whats-new.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="api-calls.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="working-with-usage-data.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="working-with-documentation.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="tools.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="event-types.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="alerts.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="time-zones.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
<xi:include href="Revision_History.xml" xmlns:xi="http://www.w3.org/2001/XInclude" />
/>
```

所有xml檔案都是DocBook格式



注意

DocBook format is well [documented](#), refer to the documentation for any questions about DocBook tags

When writing documentation, you therefore need to located the book,chapter and section of the content you want to write/correct. Or create a new book,chapter,section.

You will then learn much more about DocBook tagging. In order to write this chapter about documentation, I added the *working-with-documentation.xml* file describing a chapter in the Developer book and I created several sections within that chapter like so:

```
<chapter id="working-with-documentation">
  <title>Preparing and Building CloudStack Documentation</title>
  <para>This chapter describes how to install publican, how to write new
documentation and build a guide as well as how to build a translated version of the
documentation using transifex</para>
  <xi:include href="installing-publican.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="building-documentation.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="writing-new-documentation.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
  <xi:include href="building-translation.xml"
xmlns:xi="http://www.w3.org/2001/XInclude" />
</chapter>
```

Note the id within the chapter tag, it represents the basename of the xml file describing the chapter.

For translation purposes it is important that this basename be less than 50 characters long.

This chapter also refers to xml files which contains each section. While you could embed the sections directly in the chapter file and as a matter of fact also write the chapters within a single book file. Breaking things up in smaller files at the granularity of the section, allows us to re-use any section to build different books.

For completeness here is an example of a section:

```
<section id="building-documentation">
  <title>Building CloudStack Documentation</title>
  <para>To build a specific guide, go to the source tree of the documentation in
/docs and identify the guide you want to build.</para>
```

```

<para>Currently there are four guides plus the release notes, all defined in
publican configuration files:</para>
<programlisting>
publican-adminguide.cfg
publican-devguide.cfg
publican-installation.cfg
publican-plugin-niciranvp.cfg
publican-release-notes.cfg
</programlisting>
<para>To build the Developer guide for example, do the following:</para>
<programlisting>publican build --config=publican-devguide.cfg --formats=pdf --
langs=en-US</programlisting>
<para>A pdf file will be created in tmp/en-US/pdf, you may choose to build the
guide in a different format like html. In that case just replace the format value.</para>
</section>

```

Happy Publishing and DocBooking.

7.4. 翻譯CloudStack文件

Now that you know how to build the documentation with Publican, let's move on to building it in different languages. Publican helps us build the documentation in various languages by using Portable Object Template (POT) files and Portable Objects (PO) files for each language.

The POT files are generated by parsing all the DocBook files in the language of origin, en-US for us, and creating a long list of strings for each file that needs to be translated. The translation can be done by hand directly in the PO files of each target language or via the transifex service.



注意

Transifex is a free service to help translate documents and organize distributed teams of translators. Anyone interested in helping with the translation should get an account on Transifex. Three CloudStack projects exist on Transifex. It is recommended to tour those projects to become familiar with Transifex:

- » https://www.transifex.com/projects/p/ACS_DOCS/
- » https://www.transifex.com/projects/p/ACS_Runbook/
- » <https://www.transifex.com/projects/p/CloudStackUI/>



警告

The pot directory should already exist in the source tree. If you want to build an up to date translation, you might have to update it to include any pot file that was not previously generated. To register new resources on transifex, you will need to be an admin of the transifex CloudStack site. Send an email to the developer list if you want access.

First we need to generate the .pot files for all the DocBook xml files needed for a particular guide. This is well explained at the publican website in a section on how to [prepare](#) a document for translation.

The basic command to execute to build the pot files for the developer guide is:

```
publican update_pot --config=publican-devguide.cfg
```

This will create a pot directory with pot files in it, one for each corresponding xml files needed to build the guide. Once generated, all pots files need to be configured for translation using transifex this is best done by using the transifex client that you can install with the following command (For RHEL and its derivatives):

```
yum install transifex-client
```

The transifex client is also available via PyPi and you can install it like this:

```
easy_install transifex-client
```

Once you have installed the transifex client you can run the `setx.sh` script in the `docs` directory. This will create the `.tx/config` file used by transifex to push and pull all translation strings.

All the resource files need to be uploaded to transifex, this is done with the transifex client like so:

```
tx push -s
```

Once the translators have completed translation of the documentation, the translated strings can be pulled from transifex like so:

```
tx pull -a
```

If you wish to push specific resource files or pull specific languages translation strings, you can do so with the transifex client. A complete documentation of the client is available on the [client](#) website

When you pull new translation strings a directory will be created corresponding to the language of the translation. This directory will contain PO files that will be used by Publican to create the documentation in that specific language. For example assuming that you pull the French translation whose language code is fr-FR, you will build the documentation with publican:

```
publican build --config=publican-devguide.cfg --formats=html --langs=fr-FR
```



警告

Some languages like Chinese or Japanese will not render well in pdf format and html should be used.

7.4.1. 翻譯CloudStack文件

有兩種翻譯文件的方式：

- » 使用Transifex網站直接翻譯
- » 使用Transifex client

Once a translation is complete, a site admin will pull the translated strings within the CloudStack repository, build the documentation and publish it.

對於如何使用 Transifex網站，參見 <http://sebgoa.blogspot.ch/2012/11/translating-apache-cloudstack-docs-with.html>

對於如何使用 Transifex client，參見 <http://sebgoa.blogspot.ch/2012/12/using-transifex-client-to-translate.html>

8.1. DevCloud

- 8.1.1. DevCloud使用模式
- 8.1.2. 建立DevCloud

8.2. Marvin

- 8.2.1. 建立及安裝 Marvin

8.3. CloudMonkey

- 8.3.1. Installing CloudMonkey
- 8.3.2. 系統設定
- 8.3.3. API Discovery
- 8.3.4. Tabular Output
- 8.3.5. Interactive Shell Usage
- 8.3.6. Starting a Virtual Machine instance with CloudMonkey
- 8.3.7. Scripting with CloudMonkey

8.4. Apache Libcloud

8.1. DevCloud

DevCloud是 CloudStack 的沙盒，提供如同Virtual Box的應用，作為開發環境來輕鬆測試新的CloudStack 開發工作。此沙盒也被在提供*Cloud in a box*後，用來訓練及CloudStack 演示

注意

DevCloud由論壇成員提供，並不是官方CloudStack 釋出
CloudStack原始碼提供工具給您建立自己的DevCloud

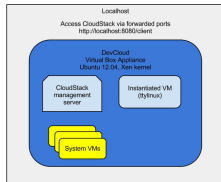
警告

DevCloud正在開發，並且應被視為Work In Progress (WIP)，Wiki是最新的文件：
<https://cwiki.apache.org/confluence/display/CLOUDSTACK/DevCloud>

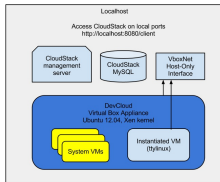
8.1.1. DevCloud使用模式

DevCloud可以就以下方式使用：

1. 完整的sandbox環境。CloudStack運行在Virtual Box的DevCloud虛擬機中。
In this mode, the CloudStack management server runs within the instance and nested virtualization allows instantiation of tiny VMs within DevCloud itself. CloudStack code modifications are done within DevCloud.
以下圖形顯示沙盒模式的架構



2. 環境部署，當CloudStack 程式已在本地主機開發，最終建構在DevCloud開發
此模式使用為測試CloudStack 4.0.0 incubating 釋出，參閱以下影片：<http://vimeo.com/54621457>
3. 離線模式，將DevCloud作為主機，CloudStack管理伺服器在本地主機執行
此模式用為離線介面，由Virtual Box 偏好設定定義，參閱以下影片：<http://vimeo.com/54610161>
以下為離線模式架構的草圖



8.1.2. 建立DevCloud

DevCloud應用程式可以從Wiki下載。<https://cwiki.apache.org/confluence/display/CLOUDSTACK/DevCloud>。也可以從scratch建立，程式正在開發，以提供替代的方法建立，DevCloud以veewee, Vagrant 及Puppet為基礎

目標為自動化DevCloud build，並相容任何CloudStack釋出的原始碼

警告

正在努力研發，程式位於tools/devcloud下的source tree
A preliminary wiki page describes the build at
<https://cwiki.apache.org/confluence/display/CLOUDSTACK/Building+DevCloud>

8.2. Marvin

Marvin是 CloudStack自動化的骨幹，源自於累積測試的工具，但現在也使用為建立DevCloud，同時也提供Python CloudStack API

注意

Marvin完整文件在Wiki <https://cwiki.apache.org/CLOUDSTACK/testing-with-python.html>
原始碼在 [tools/marvin](https://github.com/apache/cloudstack-marvin)

8.2.1. 建立及安裝 Marvin

Marvin是以Maven建立，與APIdoc相關。想要建立Marvin來使用root tree of CloudStack執行以下：

```
mvn -P developer -pl :cloud-apidoc
```

```
mvn -P developer -pl :cloud-marvin
```

If successful the build will have created the cloudstackAPI Python package under tools/marvin/marvin/cloudstackAPI as well as a gzipped Marvin package under tools/marvin dist. To install the Python Marvin module do the following in tools/marvin:

```
sudo python ./setup.py install
```

當 Python module安裝後dependencies會自動下載，並且您就可以在Python使用Marvin，檢查您可以在啟動前匯入module

```
$ python
Python 2.7.3 (default, Nov 17 2012, 19:54:34)
[GCC 4.2.1 Compatible Apple Clang 4.1 ((tags/Apple/clang-421.11.66))] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import marvin
>>> from marvin.cloudstackAPI import *
>>>
```

您也可以使用 *pip* 或 *easy_install* 來安裝，藉由使用 tools/marvin/dist 中的 local distribution package：

```
pip install tools/marvin/dist/Marvin-0.1.0.tar.gz
```

或：

```
easy_install tools/marvin/dist/Marvin-0.1.0.tar.gz
```

8.3. CloudMonkey

CloudMonkey is the CloudStack Command Line Interface (CLI). It is written in Python. CloudMonkey can be used both as an interactive shell and as a command line tool which simplifies CloudStack configuration and management. It can be used with CloudStack releases since the 4.0.x branch.



警告

CloudMonkey is still under development and should be considered a Work In Progress (WIP), the wiki is the most up to date documentation:
<https://cwiki.apache.org/CLOUDSTACK/cloudstack-cloudmonkey-cli.html>

8.3.1. Installing CloudMonkey

CloudMonkey is dependent on *readline*, *pygments*, *prettytable*, when installing from source you will need to resolve those dependencies. Using the cheese shop, the dependencies will be automatically installed.

There are three ways to get CloudMonkey. Via the official CloudStack source releases or via a community maintained distribution at [the cheese shop](https://cheeseshop.python.org/pypi/cloudmonkey/). Developers can also get it directly from the git repository in *tools/cli*.

- ▶ Via the official Apache CloudStack releases as well as the git repository.

```
$ git clone https://git-wip-us.apache.org/repos/asf/cloudstack.git # (optional if using
a release download)
$ mvn clean install -P developer
$ cd tools/cli # cloudmonkey-x.x.x.tar.gz will be built in dist
$ python setup.py build
$ python setup.py install
```

- ▶ Via a community maintained package on Cheese Shop

```
pip install cloudmonkey
```

8.3.2. 系統設定

To configure CloudMonkey you can edit the *~/.cloudmonkey/config* file in the user's home directory as shown below. The values can also be set interactively at the cloudmonkey prompt. Logs are kept in *~/.cloudmonkey/log*, and history is stored in *~/.cloudmonkey/history*. Discovered apis are listed in *~/.cloudmonkey/cache*. Only the log and history files can be custom paths and can be configured by setting appropriate file paths in *~/.cloudmonkey/config*

```
$ cat ~/.cloudmonkey/config
[core]
log_file = /Users/sebastiengoasguen/.cloudmonkey/log
asynblock = true
paramcompletion = false
history_file = /Users/sebastiengoasguen/.cloudmonkey/history

[ui]
color = true
prompt = >
tabularize = false

[user]
secretkey
=VdaACyB0LV9eNjTetIOE1cVQkvJck_J_Q1jX_FCHRj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhwJaw7FF3akA3K6Q
apikey = p1gWJfZK4gyS3m0MTvmjUVg-X-
j1w1nfaUJ9GAbBbF9EdMkAYMm1Lqzq1E1ZLq_u38zCm0bewzGUDP66mg

[server]
path = /client/api
host = localhost
protocol = http
port = 8080
timeout = 3600
```

The values can also be set at the CloudMonkey prompt. The API and secret keys are obtained via the CloudStack UI or via a raw api call.

```
$ cloudmonkey
└─ Apache CloudStack cloudmonkey 4.1.1-snapshot. Type help or ? to list commands.

> set prompt myprompt>
myprompt> set host localhost
myprompt> set port 8080
myprompt> set apikey <your api key>
```

```
myprompt> set secretkey <your secret key>
```

You can use CloudMonkey to interact with a local cloud, and even with a remote public cloud. You just need to set the host value properly and obtain the keys from the cloud administrator.

8.3.3. API Discovery

注意

In CloudStack 4.0.* releases, the list of api calls available will be pre-cached, while starting with CloudStack 4.1 releases and above an API discovery service is enabled. CloudMonkey will discover automatically the api calls available on the management server. The sync command in CloudMonkey pulls a list of apis which are accessible to your user role, along with help docs etc. and stores them in ~/.cloudmonkey/cache. This allows cloudmonkey to be adaptable to changes in mgmt server, so in case the sysadmin enables a plugin such as Nicira NVP for that user role, the users can get those changes. New verbs and grammar (DSL) rules are created on the fly.

To discover the APIs available do:

```
> sync
324 APIs discovered and cached
```

8.3.4. Tabular Output

The number of key/value pairs returned by the api calls can be large resulting in a very long output. To enable easier viewing of the output, a tabular formatting can be setup. You may enable tabular listing and even choose set of column fields, this allows you to create your own field using the filter param which takes in comma separated argument. If argument has a space, put them under double quotes. The create table will have the same sequence of field filters provided

To enable it, use the set function and create filters like so:

```
> set tabularize true
> list users filter=id, domain, account
count = 1
user:
+-----+-----+-----+
|          id          | domain | account |
+-----+-----+-----+
| 7ed6d5da-93b2-4545-a502-23d20b48ef2a | ROOT  | admin  |
+-----+-----+-----+
```

8.3.5. Interactive Shell Usage

To start learning CloudMonkey, the best is to use the interactive shell. Simply type CloudMonkey at the prompt and you should get the interactive shell.

At the CloudMonkey prompt press the tab key twice, you will see all potential verbs available. Pick one, enter a space and then press tab twice. You will see all actions available for that verb

```
cloudmonkey>
EOF
reconnect  assign      cancel      create      detach      extract      ldap        prepare
activate   restart    shell       update      disable     generate     list        query
register    associate  change     delete      enable      get          mark        quit
add         attach     configure  deploy      exit        help         migrate     reboot
remove      revoke     stop        destroy
api         authorize  copy
reset      set        suspend
cloudmonkey>create
account    diskoffering
snapshot   tags
autoscalepolicy  domain
snapshotpolicy  template
autoscalevmgroup  firewallrule
sshkeypair      user
autoscalevmprofile  instancegroup
staticroute     virtualrouteruleelement
condition       ipforwardingrule
storagenetworkiprange  vlaniprange
counter         lbstickinesspolicy
storagepool    volume
loadbalancerrule
vpc
network
networkoffering
networkacl
project
vpconnection
networkoffering
remoteaccessvpn
vpcustomergateway
physicalnetwork
securitygroup
vpngateway
pod
zone
serviceoffering
```

Picking one action and entering a space plus the tab key, you will obtain the list of parameters for that specific api call.

```
cloudmonkey>create network
account=          domainid=          isAsync=          networkdomain=    projectid=
vlan=            endip=            name=            networkofferingid= startip=
acltype=         gateway=          netmask=         physicalnetworkid=
vpcid=
displaytext=
subdomainaccess= zoneid=
```

To get additional help on that specific api call you can use the following:

```
cloudmonkey>create network -h
Creates a network
Required args: displaytext name networkofferingid zoneid
Args: account acltype displaytext domainid endip gateway isAsync name netmask networkdomain
networkofferingid physicalnetworkid projectid startip subdomainaccess vlan vpcid zoneid

cloudmonkey>create network -help
Creates a network
Required args: displaytext name networkofferingid zoneid
Args: account acltype displaytext domainid endip gateway isAsync name netmask networkdomain
networkofferingid physicalnetworkid projectid startip subdomainaccess vlan vpcid zoneid

cloudmonkey>create network --help
Creates a network
Required args: displaytext name networkofferingid zoneid
Args: account acltype displaytext domainid endip gateway isAsync name netmask networkdomain
networkofferingid physicalnetworkid projectid startip subdomainaccess vlan vpcid zoneid
cloudmonkey>
```

Note the required arguments necessary for the calls.

注意

To find out the required parameters value, using a debugger console on the CloudStack UI might be very useful. For instance using Firebug on Firefox, you can navigate the UI and check the parameters values for each call you are making as you navigate the UI.

8.3.6. Starting a Virtual Machine instance with CloudMonkey

To start a virtual machine instance we will use the `deploy virtualmachine` call.

```
cloudmonkey>deploy virtualmachine -h
Creates and automatically starts a virtual machine based on a service offering, disk
offering, and template.
Required args: serviceofferingid templateid zoneid
Args: account diskofferingid displayname domainid group hostid hypervisor ipaddress
iptonetworklist isAsync keyboard keypair name networkids projectid securitygroupids
securitygroupnames serviceofferingid size startvm templateid userdata zoneid
```

The required arguments are `serviceofferingid`, `templateid` and `zoneid`

In order to specify the template that we want to use, we can list all available templates with the following call:

```
cloudmonkey>list templates templatefilter=all
count = 2
template:
=====
domain = ROOT
domainid = 8a111e58-e155-4482-93ce-84efff3c7c77
zoneid = e1bfdfaf-3d9b-43d4-9aea-2c9f173a1ae7
displaytext = SystemVM Template (XenServer)
ostypeid = 849d7d0a-9fbc-452a-85aa-70e0a0cbc688
passwordenabled = False
id = 6d360f79-4de9-468c-82f8-a348135d298e
size = 2101252608
isready = True
templatetype = SYSTEM
zonename = devcloud
...<snipped>
```

In this snippet, I used DevCloud and only showed the beginning output of the first template, the SystemVM template

Similarly to get the `serviceofferingid` you would do:

```
cloudmonkey>list serviceofferings | grep id
id = ef2537ad-c70f-11e1-821b-0800277e749c
id = c66c2557-12a7-4b32-94f4-48837da3fa84
id = 3d8b82e5-d8e7-48d5-a554-cf853111bc50
```

Note that we can use the linux pipe as well as standard linux commands within the interactive shell. Finally we would start an instance with the following call:

```
cloudmonkey>deploy virtualmachine templateid=13ccff62-132b-4caf-b456-e8ef20cbff0e
zoneid=e1bfdfaf-3d9b-43d4-9aea-2c9f173a1ae7 serviceofferingid=ef2537ad-c70f-11e1-821b-
0800277e749c
jobprocstatus = 0
created = 2013-03-05T13:04:51-0800
cmd = com.cloud.api.commands.DeployVMCmd
userid = 7ed6d5da-93b2-4545-a502-23d20b48ef2a
jobstatus = 1
jobid = c441d894-e116-402d-aa36-fdb45adb16b7
jobresultcode = 0
jobresulttype = object
jobresult:
=====
virtualmachine:
=====
domain = ROOT
domainid = 8a111e58-e155-4482-93ce-84efff3c7c77
haenable = False
templatename = tiny Linux
...<snipped>
```

The instance would be stopped with:

```
cloudmonkey>stop virtualmachine id=7efe0377-4102-4193-bff8-c706909cc2d2
```



注意

The `ids` that you will use will differ from this example. Make sure you use the ones that corresponds to your CloudStack cloud.

8.3.7. Scripting with CloudMonkey

All previous examples use CloudMonkey via the interactive shell, however it can be used as a straightforward CLI, passing the commands to the `cloudmonkey` command like shown below.

```
$cloudmonkey list users
```

As such it can be used in shell scripts, it can received commands via stdin and its output can be parsed like any other unix commands as mentioned before.

8.4. Apache Libcloud

There are many tools available to interface with the CloudStack API. Apache Libcloud is one of those. In this section we provide a basic example of how to use Libcloud with CloudStack. It assumes that you have access to a CloudStack endpoint and that you have the API access key and secret key of a user.

To install Libcloud refer to the libcloud website. If you are familiar with Pypi simply do:

```
pip install apache-libcloud
```

You should see the following output:

```
pip install apache-libcloud
Downloading/unpacking apache-libcloud
  Downloading apache-libcloud-0.12.4.tar.bz2 (376kB): 376kB downloaded
  Running setup.py egg_info for package apache-libcloud

Installing collected packages: apache-libcloud
  Running setup.py install for apache-libcloud

Successfully installed apache-libcloud
Cleaning up...
```

You can then open a Python interactive shell, create an instance of a CloudStack driver and call the available methods via the libcloud API.

```

>>> from libcloud.compute.types import Provider
>>> from libcloud.compute.providers import get_driver
>>> Driver = get_driver(Provider.CLOUDSTACK)
>>> apikey='plgWJfZK4gyS3mOMTvmjUVg-X-jlWlnfaUJ9GAbBbf9EdM-
kAYMmAlLqzq1ELZLYq_u38zCm0bewzGUDp66mg'
>>>
secretkey='VdaACyB0LV9eNjTet10ElcVQkvJck_J_QljX_Fchrj87ZKiy0z0ty0ZsYBkoXkY9b7eq1EhWJaw7FF3a
kA3KBQ'
>>> host='http://localhost:8080'
>>> path='/client/api'
>>> conn=Driver(apikey,secretkey,secure='False',host='localhost:8080',path=path)
>>>
conn=Driver(key=apikey,secret=secretkey,secure=False,host='localhost',port='8080',path=path)
>>> conn.list_images()
[<NodeImage: id=13ccff62-132b-4caf-b456-e8ef20cbff0e, name=tiny Linux, driver=CloudStack
...>]
>>> conn.list_sizes()
[<NodeSize: id=ef2537ad-c70f-11e1-821b-0800277e749c, name=tinyOffering, ram=100 disk=0
bandwidth=0 price=0 driver=CloudStack ...>, <NodeSize: id=c66c2557-12a7-4b32-94f4-
48831da3fa84, name=Small Instance, ram=512 disk=0 bandwidth=0 price=0 driver=CloudStack
...>, <NodeSize: id=3d8b82e5-d8e7-48d5-a554-cf853111bc50, name=Medium Instance, ram=1024
disk=0 bandwidth=0 price=0 driver=CloudStack ...>]
>>> images=conn.list_images()
>>> offerings=conn.list_sizes()
>>> node=conn.create_node(name='toto',image=images[0],size=offerings[0])
>>> help(node)
>>> node.get_uuid()
'b1aa381ba1de7f2d5048e248848993d5a900984f'
>>> node.name
u'toto'

```

One of the interesting use cases of Libcloud is that you can use multiple Cloud Providers, such as AWS, Rackspace, OpenNebula, vCloud and so on. You can then create Driver instances to each of these clouds and create your own multi cloud application.

Event Types

VM.CREATE	TEMPLATE.EXTRACT	SG.REVOKE.INGRESS
VM.DESTROY	TEMPLATE.UPLOAD	HOST.RECONNECT
VM.START	TEMPLATE.CLEANUP	MAINT.CANCEL
VM.STOP	VOLUME.CREATE	MAINT.CANCEL.PS
VM.REBOOT	VOLUME.DELETE	MAINT.PREPARE
VM.UPGRADE	VOLUME.ATTACH	MAINT.PREPARE.PS
VM.RESETPASSWORD	VOLUME.DETACH	VPN.REMOTE.ACCESS.CREATE
ROUTER.CREATE	VOLUME.UPLOAD	VPN.USER.ADD
ROUTER.DESTROY	SERVICEOFFERING.CREATE	VPN.USER.REMOVE
ROUTER.START	SERVICEOFFERING.UPDATE	NETWORK.RESTART
ROUTER.STOP	SERVICEOFFERING.DELETE	UPLOAD.CUSTOM.CERTIFICATE
ROUTER.REBOOT	DOMAIN.CREATE	UPLOAD.CUSTOM.CERTIFICATE
ROUTER.HA	DOMAIN.DELETE	STATICNAT.DISABLE
PROXY.CREATE	DOMAIN.UPDATE	SSVM.CREATE
PROXY.DESTROY	SNAPSHOT.CREATE	SSVM.DESTROY
PROXY.START	SNAPSHOT.DELETE	SSVM.START
PROXY.STOP	SNAPSHOTPOLICY.CREATE	SSVM.STOP
PROXY.REBOOT	SNAPSHOTPOLICY.UPDATE	SSVM.REBOOT
PROXY.HA	SNAPSHOTPOLICY.DELETE	SSVM.H
VNC.CONNECT	VNC.DISCONNECT	NET.IP.ASSIGN
NET.IP.RELEASE	NET.RULEADD	NET.RULEDELETE
NET.RULEMODIFY	NETWORK.CREATE	NETWORK.DELETE
LB.ASSIGN.TO.RULE	LB.REMOVE.FROM.RULE	LB.CREATE
LB.DELETE	LB.UPDATE	USER.LOGIN
USER.LOGOUT	USER.CREATE	USER.DELETE
USER.UPDATE	USER.DISABLE	TEMPLATE.CREATE
TEMPLATE.DELETE	TEMPLATE.UPDATE	TEMPLATE.COPY
TEMPLATE.DOWNLOAD.START	TEMPLATE.DOWNLOAD.SUCCESS	TEMPLATE.DOWNLOAD.FAILED
ISO.CREATE	ISO.DELETE	ISO.COPY
ISO.ATTACH	ISO.DETACH	ISO.EXTRACT
ISO.UPLOAD	SERVICE.OFFERING.CREATE	SERVICE.OFFERING.EDIT
SERVICE.OFFERING.DELETE	DISK.OFFERING.CREATE	DISK.OFFERING.EDIT
DISK.OFFERING.DELETE	NETWORK.OFFERING.CREATE	NETWORK.OFFERING.EDIT
NETWORK.OFFERING.DELETE	POD.CREATE	POD.EDIT
POD.DELETE	ZONE.CREATE	ZONE.EDIT
ZONE.DELETE	VLAN.IP.RANGE.CREATE	VLAN.IP.RANGE.DELETE
CONFIGURATION.VALUE.EDIT	SG.AUTH.INGRESS	

警告

The following is the list of alert type numbers. The current alerts can be found by calling listAlerts.

MEMORY = 0

CPU = 1

STORAGE =2

STORAGE_ALLOCATED = 3

PUBLIC_IP = 4

PRIVATE_IP = 5

HOST = 6

USERVM = 7

DOMAIN_ROUTER = 8

CONSOLE_PROXY = 9

ROUTING = 10// lost connection to default route (to the gateway)

STORAGE_MISC = 11 // lost connection to default route (to the gateway)

USAGE_SERVER = 12 // lost connection to default route (to the gateway)

MANAGMENT_NODE = 13 // lost connection to default route (to the gateway)

DOMAIN_ROUTER_MIGRATE = 14

CONSOLE_PROXY_MIGRATE = 15

USERVM_MIGRATE = 16

VLAN = 17

SSVM = 18

USAGE_SERVER_RESULT = 19

STORAGE_DELETE = 20;

UPDATE_RESOURCE_COUNT = 21; //Generated when we fail to update the resource count

USAGE_SANITY_RESULT = 22;

DIRECT_ATTACHED_PUBLIC_IP = 23;

LOCAL_STORAGE = 24;

RESOURCE_LIMIT_EXCEEDED = 25; //Generated when the resource limit exceeds the limit.
Currently used for recurring snapshots only

時區

CloudStack可使用以下時區確認器，許多時區有要求或選擇的欄位，這些包含在Configuration表中排程常用快取物件、建立使用者及指定使用時區

Etc/GMT+12	Etc/GMT+11	Pacific/Samoa
Pacific/Honolulu	US/Alaska	America/Los_Angeles
Mexico/BajaNorte	US/Arizona	US/Mountain
America/Chihuahua	America/Chicago	America/Costa_Rica
America/Mexico_City	Canada/Saskatchewan	America/Bogota
America/New_York	America/Caracas	America/Asuncion
America/Cuiaba	America/Halifax	America/La_Paz
America/Santiago	America/St_Johns	America/Araguaina
America/Argentina/Buenos_Aires	America/Cayenne	America/Godthab
America/Montevideo	Etc/GMT+2	Atlantic/Azores
Atlantic/Cape_Verde	Africa/Casablanca	Etc/UTC
Atlantic/Reykjavik	Europe/London	CET
Europe/Bucharest	Africa/Johannesburg	Asia/Beirut
Africa/Cairo	Asia/Jerusalem	Europe/Minsk
Europe/Moscow	Africa/Nairobi	Asia/Karachi
Asia/Kolkata	Asia/Bangkok	Asia/Shanghai
Asia/Kuala_Lumpur	Australia/Perth	Asia/Taipei
Asia/Tokyo	Asia/Seoul	Australia/Adelaide
Australia/Darwin	Australia/Brisbane	Australia/Canberra
Pacific/Guam	Pacific/Auckland	

修訂記錄

修訂 0-0
初始版本

Tue May 29 2012

Jessica Tomechak