

Apache CloudStack 4.0.2

CloudStack API Developer's Guide



Apache CloudStack

Apache CloudStack 4.0.2 CloudStack API Developer's Guide

Author

Apache CloudStack

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.

Apache CloudStack is an effort undergoing incubation at The Apache Software Foundation (ASF).

Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects. While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF.

How to integrate with CloudStack using the CloudStack API.

1. Concepts	1
1.1. What Is CloudStack?	1
1.2. What Can CloudStack Do?	1
1.3. Deployment Architecture Overview	2
1.3.1. Management Server Overview	3
1.3.2. Cloud Infrastructure Overview	3
1.3.3. Networking Overview	4
2. Introduction for Developers	5
2.1. Roles	5
2.2. API Reference Documentation	5
2.3. Getting Started	5
3. What's New in the API?	7
3.1. What's New in the API for 4.0	7
3.1.1. Changed API Commands in 4.0.0-incubating	7
3.1.2. Added API Commands in 4.0.0-incubating	10
3.2. What's New in the API for 3.0	11
3.2.1. Enabling Port 8096	11
3.2.2. Stopped VM	12
3.2.3. Change to Behavior of List Commands	12
3.2.4. Removed API commands	13
3.2.5. Added API commands in 3.0	13
3.2.6. Added CloudStack Error Codes	15
4. Calling the CloudStack API	17
4.1. Making API Requests	17
4.2. Enabling API Call Expiration	17
4.3. Signing API Requests	18
4.4. Responses	19
4.4.1. Response Formats: XML and JSON	19
4.4.2. Maximum Result Pages Returned	20
4.4.3. Error Handling	20
4.5. Asynchronous Commands	21
4.5.1. Job Status	21
4.5.2. Example	21
5. Working With Usage Data	25
5.1. Usage Record Format	25
5.1.1. Virtual Machine Usage Record Format	25
5.1.2. Network Usage Record Format	25
5.1.3. IP Address Usage Record Format	26
5.1.4. Disk Volume Usage Record Format	26
5.1.5. Template, ISO, and Snapshot Usage Record Format	27
5.1.6. Load Balancer Policy or Port Forwarding Rule Usage Record Format	27
5.1.7. Network Offering Usage Record Format	28
5.1.8. VPN User Usage Record Format	28
5.2. Usage Types	29
5.3. Example response from listUsageRecords	30
5.4. Dates in the Usage Record	31
A. Event Types	33
B. Alerts	35
C. Time Zones	37
D. Revision History	39

Concepts

1.1. What Is CloudStack?

CloudStack is an open source software platform that pools computing resources to build public, private, and hybrid Infrastructure as a Service (IaaS) clouds. CloudStack manages the network, storage, and compute nodes that make up a cloud infrastructure. Use CloudStack to deploy, manage, and configure cloud computing environments.

Typical users are service providers and enterprises. With CloudStack, you can:

- Set up an on-demand, elastic cloud computing service. Service providers can sell self service virtual machine instances, storage volumes, and networking configurations over the Internet.
- Set up an on-premise private cloud for use by employees. Rather than managing virtual machines in the same way as physical machines, with CloudStack an enterprise can offer self-service virtual machines to users without involving IT departments.



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](http://incubator.apache.org/cloudstack/docs)¹ and [Apache CloudStack API Reference](http://incubator.apache.org/cloudstack/docs/api)² 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.

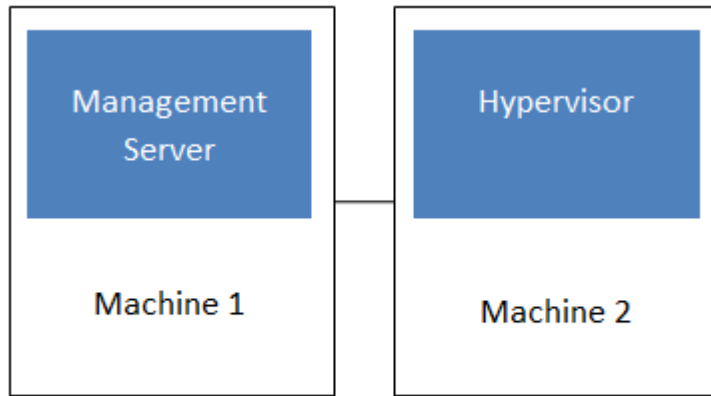
1.3. Deployment Architecture Overview

A CloudStack installation consists of two parts: the Management Server and the cloud infrastructure that it manages. When you set up and manage a CloudStack cloud, you provision resources such as hosts, storage devices, and IP addresses into the Management Server, and the Management Server manages those resources.

The minimum production installation consists of one machine running the CloudStack Management Server and another machine to act as the cloud infrastructure (in this case, a very simple infrastructure consisting of one host running hypervisor software). In its smallest deployment, a single machine can act as both the Management Server and the hypervisor host (using the KVM hypervisor).

¹ <http://incubator.apache.org/cloudstack/docs>

² <http://incubator.apache.org/cloudstack/docs/api>



Simplified view of a basic deployment

A more full-featured installation consists of a highly-available multi-node Management Server installation and up to tens of thousands of hosts using any of several advanced networking setups. For information about deployment options, see [Choosing a Deployment Architecture](#).

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:

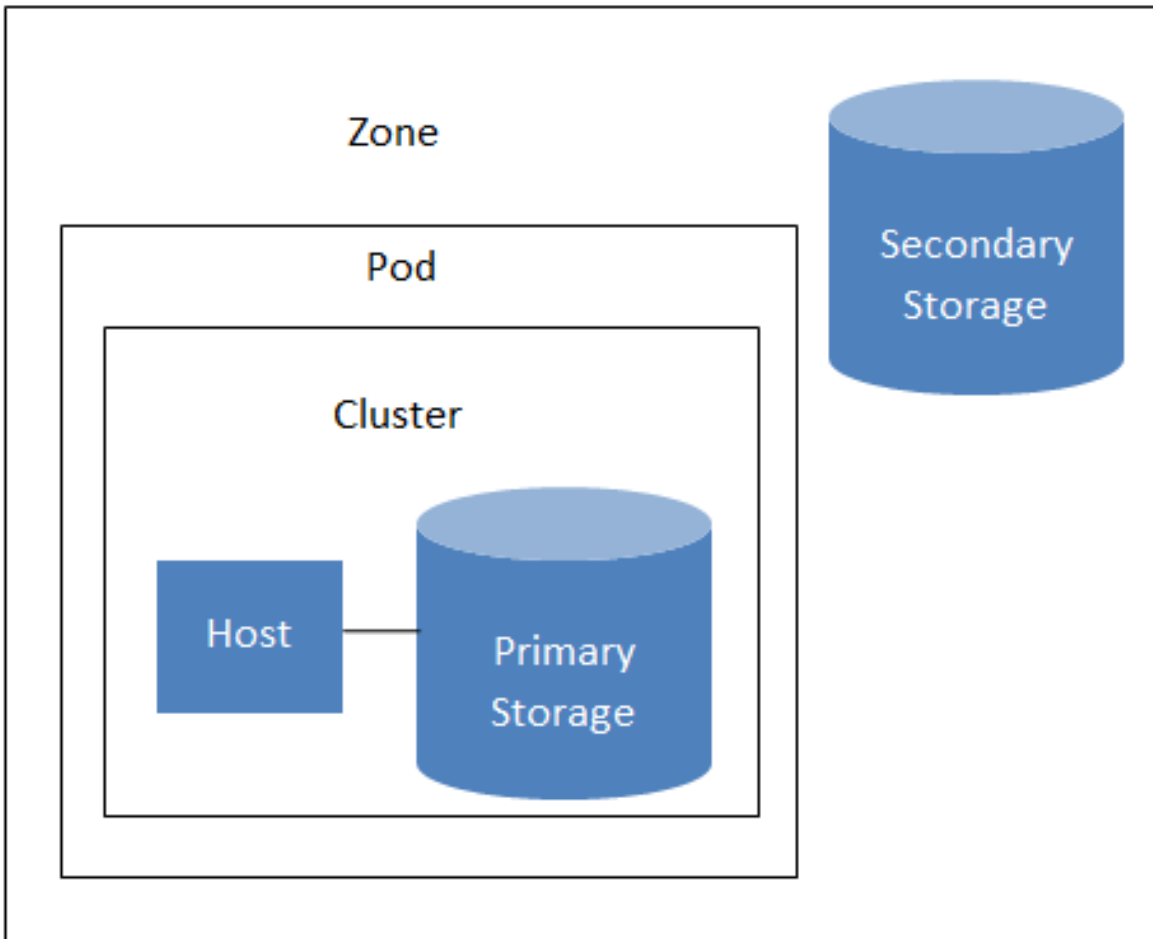
- Provides the web user interface for the administrator and a reference user interface for end users.
- Provides the APIs for CloudStack.
- Manages the assignment of guest VMs to particular hosts.
- Manages the assignment of public and private IP addresses to particular accounts.
- Manages the allocation 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. Cloud Infrastructure Overview

The Management Server manages one or more zones (typically, datacenters) containing host computers where guest virtual machines will run. The cloud infrastructure is organized as follows:

- Zone: Typically, a zone is equivalent to a single datacenter. A zone consists of one or more pods and secondary storage.
- Pod: A pod is usually one rack of hardware that includes a layer-2 switch and one or more clusters.
- Cluster: A cluster consists of one or more hosts and primary storage.

- Host: A single compute node within a cluster. The hosts are where the actual cloud services run in the form of guest virtual machines.
- Primary storage is associated with a cluster, and it stores the disk volumes for all the VMs running on hosts in that cluster.
- Secondary storage is associated with a zone, and it stores templates, ISO images, and disk volume snapshots.



Nested organization of a zone

More Information

For more information, see documentation on cloud infrastructure concepts.

1.3.3. Networking Overview

CloudStack offers two types of networking scenario:

- Basic. For AWS-style networking. Provides a single network where guest isolation can be provided through layer-3 means such as security groups (IP address source filtering).
- Advanced. For more sophisticated network topologies. This network model provides the most flexibility in defining guest networks.

For more details, see Network Setup.

Introduction for Developers

2.1. 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.

2.2. API Reference Documentation

2.2 API Reference:

<http://download.cloud.com/support/downloads.html/>

3.0 API Reference:

http://download.cloud.com/releases/3.0.0/api_3.0.0/TOC_Root_Admin.html/

2.3. 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.

What's New in the API?

The following describes any new major features of each CloudStack version as it applies to API usage.

3.1. What's New in the API for 4.0

3.1.1. Changed API Commands in 4.0.0-incubating

API Commands	Description
copyTemplate	The commands in this list have a single new response parameter, and no other changes.
prepareTemplate	
registerTemplate	New response parameter: tags(*)
updateTemplate	 Note Many other commands also have the new tags(*) parameter in addition to other changes; those commands are listed separately.
createProject	
activateProject	
suspendProject	
updateProject	
listProjectAccounts	
createVolume	
migrateVolume	
attachVolume	
detachVolume	
uploadVolume	
createSecurityGroup	
registerIso	
copyIso	
updateIso	
createIpForwardingRule	
listIpForwardingRules	
createLoadBalancerRule	
updateLoadBalancerRule	
createSnapshot	

API Commands	Description
rebootVirtualMachine attachIso detachIso listLoadBalancerRuleInstances resetPasswordForVirtualMachine changeServiceForVirtualMachine recoverVirtualMachine startVirtualMachine migrateVirtualMachine deployVirtualMachine assignVirtualMachine updateVirtualMachine restoreVirtualMachine stopVirtualMachine destroyVirtualMachine	The commands in this list have two new response parameters, and no other changes. New response parameters: keypair, tags(*)
listSecurityGroups listFirewallRules listPortForwardingRules listSnapshots listIos listProjects listTemplates listLoadBalancerRules	The commands in this list have the following new parameters, and no other changes. New request parameter: tags (optional) New response parameter: tags(*)
listF5LoadBalancerNetworks listNetscalerLoadBalancerNetworks listSrxFirewallNetworks updateNetwork	The commands in this list have three new response parameters, and no other changes. New response parameters: canusefordeploy, vpcid, tags(*)
createZone updateZone	The commands in this list have the following new parameters, and no other changes. New request parameter: localstorageenabled (optional) New response parameter: localstorageenabled

API Commands	Description
listZones	New response parameter: localstorageenabled
rebootRouter changeServiceForRouter startRouter destroyRouter stopRouter	The commands in this list have two new response parameters, and no other changes. New response parameters: vpcid, nic(*)
updateAccount disableAccount listAccounts markDefaultZoneForAccount enableAccount	The commands in this list have three new response parameters, and no other changes. New response parameters: vpcavailable, vplimit, vptotal
listRouters	New request parameters: forvpc (optional), vpcid (optional) New response parameters: vpcid, nic(*)
listNetworkOfferings	New request parameters: forvpc (optional) New response parameters: forvpc
listVolumes	New request parameters: details (optional), tags (optional) New response parameters: tags(*)
addTrafficMonitor	New request parameters: excludezones (optional), includezones (optional)
createNetwork	New request parameters: vpcid (optional) New response parameters: canusefordeploy, vpcid, tags(*)
listPublicIpsAddresses	New request parameters: tags (optional), vpcid (optional) New response parameters: vpcid, tags(*)
listNetworks	New request parameters: canusefordeploy (optional), forvpc (optional), tags (optional), vpcid (optional) New response parameters: canusefordeploy, vpcid, tags(*)
restartNetwork	New response parameters: vpcid, tags(*)
enableStaticNat	New request parameter: networkid (optional)
createDiskOffering	New request parameter: storagetype (optional) New response parameter: storagetype
listDiskOfferings	New response parameter: storagetype
updateDiskOffering	New response parameter: storagetype
createFirewallRule	Changed request parameters: ipaddressid (old version - optional, new version - required)

API Commands	Description
	New response parameter: tags(*)
listVirtualMachines	New request parameters: isoid (optional), tags (optional), templateid (optional) New response parameters: keypair, tags(*)
updateStorageNetworkIpRange	New response parameters: id, endip, gateway, netmask, networkid, podid, startip, vlan, zoneid

3.1.2. Added API Commands in 4.0.0-incubating

- createCounter (Adds metric counter)
- deleteCounter (Deletes a counter)
- listCounters (List the counters)
- createCondition (Creates a condition)
- deleteCondition (Removes a condition)
- listConditions (List Conditions for the specific user)
- createTags. Add tags to one or more resources. Example:

```
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. Remove tags from one or more resources. Example:

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

- listTags (Show currently defined resource tags)
- createVPC (Creates a VPC)
- listVPCs (Lists VPCs)
- deleteVPC (Deletes a VPC)
- updateVPC (Updates a VPC)
- restartVPC (Restarts a VPC)
- createVPCOffering (Creates VPC offering)
- updateVPCOffering (Updates VPC offering)
- deleteVPCOffering (Deletes VPC offering)

- listVPCOfferings (Lists VPC offerings)
- createPrivateGateway (Creates a private gateway)
- listPrivateGateways (List private gateways)
- deletePrivateGateway (Deletes a Private gateway)
- createNetworkACL (Creates a ACL rule the given network (the network has to belong to VPC))
- deleteNetworkACL (Deletes a Network ACL)
- listNetworkACLs (Lists all network ACLs)
- createStaticRoute (Creates a static route)
- deleteStaticRoute (Deletes a static route)
- listStaticRoutes (Lists all static routes)
- createVpnCustomerGateway (Creates site to site vpn customer gateway)
- createVpnGateway (Creates site to site vpn local gateway)
- createVpnConnection (Create site to site vpn connection)
- deleteVpnCustomerGateway (Delete site to site vpn customer gateway)
- deleteVpnGateway (Delete site to site vpn gateway)
- deleteVpnConnection (Delete site to site vpn connection)
- updateVpnCustomerGateway (Update site to site vpn customer gateway)
- resetVpnConnection (Reset site to site vpn connection)
- listVpnCustomerGateways (Lists site to site vpn customer gateways)
- listVpnGateways (Lists site 2 site vpn gateways)
- listVpnConnections (Lists site to site vpn connection gateways)
- enableCiscoNexusVSM (Enables Nexus 1000v dvSwitch in CloudStack.)
- disableCiscoNexusVSM (Disables Nexus 1000v dvSwitch in CloudStack.)
- deleteCiscoNexusVSM (Deletes Nexus 1000v dvSwitch in CloudStack.)
- listCiscoNexusVSMs (Lists the control VLAN ID, packet VLAN ID, and data VLAN ID, as well as the IP address of the Nexus 1000v dvSwitch.)

3.2. What's New in the API for 3.0

3.2.1. Enabling Port 8096

Port 8096, which allows API calls without authentication, is closed and disabled by default on any fresh 3.0.1 installations. You can enable 8096 (or another port) for this purpose as follows:

1. Ensure that the first Management Server is installed and running.

2. Set the global configuration parameter `integration.api.port` to the desired port.
3. Restart the Management Server.
4. On the Management Server host machine, create an iptables rule allowing access to that port.

3.2.2. Stopped VM

CloudStack now supports creating a VM without starting it. You can determine whether the VM needs to be started as part of the VM deployment. A VM can now be deployed in two ways: create and start a VM (the default method); or create a VM and leave it in the stopped state.

A new request parameter, `startVM`, is introduced in the `deployVm` API to support the stopped VM feature.

The possible values are:

- `true` - The VM starts as a part of the VM deployment.
- `false` - The VM is left in the stopped state at the end of the VM deployment.

The default value is `true`.

3.2.3. 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	Description
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.

3.2.4. Removed API commands

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

3.2.5. Added API commands in 3.0

3.2.5.1. 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.

3.2.5.2. 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.

3.2.5.3. 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.)	IdapConfig (Configure the LDAP context for this site.)
addSwift (Adds Swift.)	listSwifts (List Swift.)	migrateVolume (Migrate volume)

Chapter 3. What's New in the API?

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 serviceProvider to a physical network)	deleteNetworkServiceProvider (Deletes a Network Service Provider.)	listNetworkServiceProviders (Lists network serviceproviders for a given physical network.)
updateNetworkServiceProvider (Updates a network serviceProvider 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.)		

3.2.6. Added CloudStack Error Codes

You can now find the CloudStack-specific error code in the exception response for each type of exception. The following list of error codes is added to the new class named CSExceptionErrorCode. These codes are applicable in CloudStack 3.0.3 and later versions.

4250 : "com.cloud.utils.exception.CloudRuntimeException"	4255 : "com.cloud.utils.exception.CloudRuntimeException"	4260 : "com.cloud.utils.exception.ExecutionException"
4265 : "com.cloud.utils.exception.HypervisorVersionChangeException"	4270 : "com.cloud.utils.exception.RuntimeCloudException"	4275 : "com.cloud.exception.CloudException"
4280 : "com.cloud.exception.AccountLimitExceeded"	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.ManagementServiceException"	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.AccountLimitExceeded"	4405 : "com.cloud.exception.AgentUnavailableException"	4410 : "com.cloud.exception.CloudAuthenticationException"
4415 : "com.cloud.exception.CloudExecutionException"	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"

Chapter 3. What's New in the API?

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.InvalidParameterException"
4490 : "com.cloud.exception.InvalidParameterValueException"	4495 : "com.cloud.exception.ManagementServiceException"	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.VirtualMachineMigrationApiException"	9999 : "com.cloud.exception.VirtualMachineMigrationApiException"	

Calling the CloudStack API

4.1. 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

A sample API GET request looks like the following:

```
http://localhost:8080/client/api?
command=deployVirtualMachine&serviceOfferingId=1&diskOfferingId=1&templateId=2&zoneId=4&apiKey=miVr6X7u6l
jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU
%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_sdah0BpjNejPgEst35eXqjB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bcikwFQ
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.

4.2. 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.

For example:

```
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=miVr6X7u6bN_sdah0BpjNejPgEst35eXq-  
jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU  
%3D
```

4.3. 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&apiKey=miVr6X7u6bN_sdah  
jB8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU  
%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.



Note

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_sdahobpjnejpgest35exq-jb8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ
```

- **Signature:** This is the hashed signature of the Base URL 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.



Note

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_sdahobpjnejpgest35exq-jb8cg20yi3yaxxcgpyuaairmfi_ejtvwz0nukkjbpmy3y2bcikwFQ&command=deployvirtualmachine&diskofferingid=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_sdahobpjnejpgest35exq-jb8CG20YI3yaxXcgpyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU%3D
```

4.4. Responses

4.4.1. 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.

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"
      }
    ]
  }
}
```

4.4.2. 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.

4.4.3. 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.

4.5. 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.

4.5.1. Job Status

The key to using an asynchronous command is the job ID that is returned immediately once the command has been executed. With the job ID, you can periodically check the job status by making calls to `queryAsyncJobResult` command. The command will return three possible job status integer values:

- 0 - Job is still in progress. Continue to periodically poll for any status changes.
- 1 - Job has successfully completed. The job will return any successful response values associated with command that was originally executed.
- 2 - Job has failed to complete. Please check the "jobresultcode" tag for failure reason code and "jobresult" for the failure reason.

4.5.2. 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>
```

Chapter 4. Calling the CloudStack API

```
<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>
```


Working With Usage Data

The Usage Server provides aggregated usage records which you can use to create billing integration for the CloudStack platform. The Usage Server works by taking data from the events log and creating summary usage records that you can access using the `listUsageRecords` API call.

The usage records show the amount of resources, such as VM run time or template storage space, consumed by guest instances. In the special case of bare metal instances, no template storage resources are consumed, but records showing zero usage are still included in the Usage Server's output.

The Usage Server runs at least once per day. It can be configured to run multiple times per day. Its behavior is controlled by configuration settings as described in the CloudStack Administration Guide.

5.1. Usage Record Format

5.1.1. Virtual Machine Usage Record Format

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

- `account` – name of the account
- `accountid` – ID of the account
- `domainid` – ID of the domain in which this account resides
- `zoneid` – Zone where the usage occurred
- `description` – A string describing what the usage record is tracking
- `usage` – String representation of the usage, including the units of usage (e.g. 'Hrs' for VM running time)
- `usagetype` – A number representing the usage type (see Usage Types)
- `rawusage` – A number representing the actual usage in hours
- `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` – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.2. Network Usage Record Format

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

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – Device ID (virtual router ID or external device ID)
- type – Device type (domain router, external load balancer, etc.)
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.3. IP Address Usage Record Format

For IP address usage the following fields exist in a usage record.

- account - name of the account
- accountid - ID of the account
- domainid - ID of the domain in which this account resides
- zoneid - Zone where the usage occurred
- description - A string describing what the usage record is tracking
- usage - String representation of the usage, including the units of usage
- usagetype - A number representing the usage type (see Usage Types)
- rawusage - A number representing the actual usage in hours
- usageid - IP address ID
- startdate, enddate - The range of time for which the usage is aggregated; see Dates in the Usage Record
- issourcenat - Whether source NAT is enabled for the IP address
- iselastic - True if the IP address is elastic.

5.1.4. Disk Volume Usage Record Format

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

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides

- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – The volume ID
- offeringid – The ID of the disk offering
- type – Hypervisor
- templateid – ROOT template ID
- size – The amount of storage allocated
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.5. Template, ISO, and Snapshot Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – The ID of the the template, ISO, or snapshot
- offeringid – The ID of the disk offering
- templateid – – Included only for templates (usage type 7). Source template ID.
- size – Size of the template, ISO, or snapshot
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.6. Load Balancer Policy or Port Forwarding Rule Usage Record Format

- account - name of the account
- accountid - ID of the account

- domainid - ID of the domain in which this account resides
- zoneid - Zone where the usage occurred
- description - A string describing what the usage record is tracking
- usage - String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype - A number representing the usage type (see Usage Types)
- rawusage - A number representing the actual usage in hours
- usageid - ID of the load balancer policy or port forwarding rule
- usagetype - A number representing the usage type (see Usage Types)
- startdate, enddate - The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.7. Network Offering Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred
- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – ID of the network offering
- usagetype – A number representing the usage type (see Usage Types)
- offeringid – Network offering ID
- virtualMachineld – The ID of the virtual machine
- virtualMachineld – The ID of the virtual machine
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.1.8. VPN User Usage Record Format

- account – name of the account
- accountid – ID of the account
- domainid – ID of the domain in which this account resides
- zoneid – Zone where the usage occurred

- description – A string describing what the usage record is tracking
- usage – String representation of the usage, including the units of usage (e.g. 'Hrs' for hours)
- usagetype – A number representing the usage type (see Usage Types)
- rawusage – A number representing the actual usage in hours
- usageid – VPN user ID
- usagetype – A number representing the usage type (see Usage Types)
- startdate, enddate – The range of time for which the usage is aggregated; see Dates in the Usage Record

5.2. Usage Types

The following table shows all usage types.

Type ID	Type Name	Description
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

Type ID	Type Name	Description
		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.

5.3. 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:

```

<listusagerecordsresponse>
  <count>1816</count>
  <usagerecord>
    <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>
</usagerecord>

... (1,815 more usage records)
</listusagerecordsresponse>
```

5.4. 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 be 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.

Appendix A. 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.IPASSIGN
NET.IPRELEASE	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	

Appendix B. Alerts

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

Appendix B. Alerts

```
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
```

Appendix C. Time Zones

The following time zone identifiers are accepted by CloudStack. There are several places that have a time zone as a required or optional parameter. These include scheduling recurring snapshots, creating a user, and specifying the usage time zone in the Configuration table.

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	

Appendix D. Revision History

Revision 0-0 **Tue May 29 2012**

Jessica Tomechak

Initial creation of book by publican
