

CloudPlatform (powered by Apache CloudStack) Version 4.2 Developer's Guide

Revised September 23, 2013 2:45 pm Pacific



Citrix CloudPlatform

CloudPlatform (powered by Apache CloudStack) Version 4.2 Developer's Guide

Revised September 23, 2013 2:45 pm Pacific

Author

Citrix CloudPlatform

© 2013 Citrix Systems, Inc. All rights reserved. Specifications are subject to change without notice. Citrix Systems, Inc., the Citrix logo, Citrix XenServer, Citrix XenCenter, and CloudPlatform are trademarks or registered trademarks of Citrix Systems, Inc. All other brands or products are trademarks or registered trademarks of their respective holders.

This guide shows how to interact programmatically with CloudPlatform. It explains how to use the CloudPlatform API to run the cloud, integrate with CloudPlatform, and access resource usage data.

1. Getting More Information and Help	1
1.1. Additional Documentation Available	1
1.2. Citrix Knowledge Center	1
1.3. Contacting Support	1
2. Introduction for Developers	3
2.1. Roles	3
2.2. Getting Started	3
3. Concepts	5
3.1. What Is CloudPlatform?	5
3.2. What Can CloudPlatform Do?	5
3.3. Deployment Architecture Overview	6
3.3.1. Management Server Overview	7
3.3.2. Cloud Infrastructure Overview	7
3.3.3. Networking Overview	8
4. What's New	11
4.1. What's New in the API for 4.2	11
4.1.1. Added API Commands in 4.2	11
4.1.2. Changed API Commands in 4.2	16
4.1.3. Deprecated APIs	29
4.2. What's New in the API for 3.0.x?	29
4.2.1. IPv6 Support in CloudPlatform	30
4.2.2. Additional VMX Settings	32
4.2.3. Resetting SSH Keys to Access VMs	33
4.2.4. XenServer Hotfix Notification	33
4.2.5. Enabling Port 8096	33
4.2.6. Stopped VM	33
4.2.7. Change to Behavior of List Commands	34
4.2.8. Added API commands	35
4.2.9. Changed API Commands	39
4.2.10. Removed API commands	44
4.2.11. Added CloudPlatform Error Codes	44
5. Making API Requests	47
5.1. Signing API Requests	47
5.2. Enabling API Call Expiration	49
5.3. Limiting the Rate of API Requests	49
5.3.1. Configuring the API Request Rate	49
5.3.2. Limitations on API Throttling	50
6. Responses	51
6.1. Response Formats: XML and JSON	51
6.2. Maximum Result Pages Returned	51
6.3. Error Handling	52
7. Asynchronous Commands	53
7.1. Job Status	53
7.2. Example	53
8. Working With Usage Data	57
8.1. Usage Record Format	57
8.1.1. Virtual Machine Usage Record Format	57
8.1.2. Network Usage Record Format	57
8.1.3. IP Address Usage Record Format	58
8.1.4. Disk Volume Usage Record Format	58

- 8.1.5. Template, ISO, and Snapshot Usage Record Format 59
- 8.1.6. Load Balancer Policy or Port Forwarding Rule Usage Record Format 59
- 8.1.7. Network Offering Usage Record Format 60
- 8.1.8. VPN User Usage Record Format 60
- 8.2. Usage Types 61
- 8.3. Example response from listUsageRecords 62
- 8.4. Dates in the Usage Record 63
- 9. Third-Party UI Plugin Framework 65**
 - 9.1. How to Write a Plugin: Overview 65
 - 9.2. How to Write a Plugin: Implementation Details 67
- 10. Writing a Storage Plugin 73**
 - 10.1. Overview of How to Write a Storage Plugin 73
 - 10.2. Implementing DataStoreDriver 74
 - 10.3. Implementing DataStoreLifecycle 74
 - 10.4. Implementing DataStoreProvider 74
 - 10.5. Place the .jar File in the Right Directory 75
 - 10.6. Edit Configuration Files 75
- 11. Amazon Web Service Interface 77**
 - 11.1. Amazon Web Services EC2 Compatible Interface 77
 - 11.2. System Requirements 77
 - 11.3. Enabling the AWS API Compatible Interface 77
 - 11.4. AWS API User Setup Steps (SOAP Only) 78
 - 11.4.1. AWS API User Registration 78
 - 11.4.2. AWS API Command-Line Tools Setup 79
 - 11.5. Supported AWS API Calls 79
- A. Event Types 83**
- B. Alerts 85**

Getting More Information and Help

1.1. Additional Documentation Available

The following guides are available:

- **Installation Guide** — Covers initial installation of CloudPlatform. It aims to cover in full detail all the steps and requirements to obtain a functioning cloud deployment.

At times, this guide mentions additional topics in the context of installation tasks, but does not give full details on every topic. Additional details on many of these topics can be found in the CloudPlatform Administration Guide. For example, security groups, firewall and load balancing rules, IP address allocation, and virtual routers are covered in more detail in the Administration Guide.

- **Administration Guide** — Discusses how to set up services for the end users of your cloud. Also covers ongoing runtime management and maintenance. This guide discusses topics like domains, accounts, service offerings, projects, guest networks, administrator alerts, virtual machines, storage, and measuring resource usage.
- **Developer's Guide** — How to use the API to interact with CloudPlatform programmatically.

1.2. Citrix Knowledge Center

Troubleshooting articles by the Citrix support team are available in the Citrix Knowledge Center, at support.citrix.com/product/cs/¹.

1.3. Contacting Support

The support team is available to help customers plan and execute their installations. To contact the support team, log in to the support portal at support.citrix.com/cloudsupport² by using the account credentials you received when you purchased your support contract.

¹ <http://support.citrix.com/product/cs/>

² <http://support.citrix.com/cloudsupport>

Introduction for Developers

2.1. Roles

The CloudPlatform 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. 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.

Concepts

3.1. What Is CloudPlatform?

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

Typical users are service providers and enterprises. With CloudPlatform, 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 CloudPlatform an enterprise can offer self-service virtual machines to users without involving IT departments.



3.2. What Can CloudPlatform Do?

Multiple Hypervisor Support

CloudPlatform works with a variety of hypervisors. A single cloud deployment can contain multiple hypervisor implementations. You have the complete freedom to choose the right hypervisor for your workload.

CloudPlatform is designed to work with open source Xen and KVM hypervisors as well as enterprise-grade hypervisors such as Citrix XenServer, VMware vSphere, and Oracle VM (OVM).

Massively Scalable Infrastructure Management

CloudPlatform 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

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

CloudPlatform 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

CloudPlatform 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

CloudPlatform provides an API that gives programmatic access to all the management features available in the UI. This API enables the creation of command line tools and new user interfaces to suit particular needs.

The CloudPlatform pluggable allocation architecture allows the creation of new types of allocators for the selection of storage and hosts.

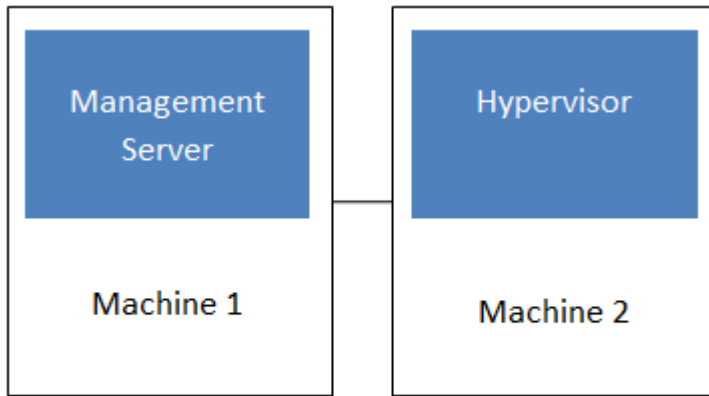
High Availability

CloudPlatform has a number of features to increase the availability of the system. The Management Server itself, which is the main controlling software at the heart of CloudPlatform, 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, CloudPlatform supports NIC bonding and the use of separate networks for storage as well as iSCSI Multipath.

3.3. Deployment Architecture Overview

A CloudPlatform installation consists of two parts: the Management Server and the cloud infrastructure that it manages. When you set up and manage a CloudPlatform 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 CloudPlatform 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 a trial installation, a single machine can act as both the Management Server and the hypervisor host (using the KVM hypervisor).



Simplified view of a basic deployment

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

3.3.1. Management Server Overview

The Management Server is the CloudPlatform 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 uses a MySQL database for persistence.

The machine where the Management Server runs must meet the system requirements described in [Minimum System Requirements](#) in the [Installation Guide](#).

The Management Server:

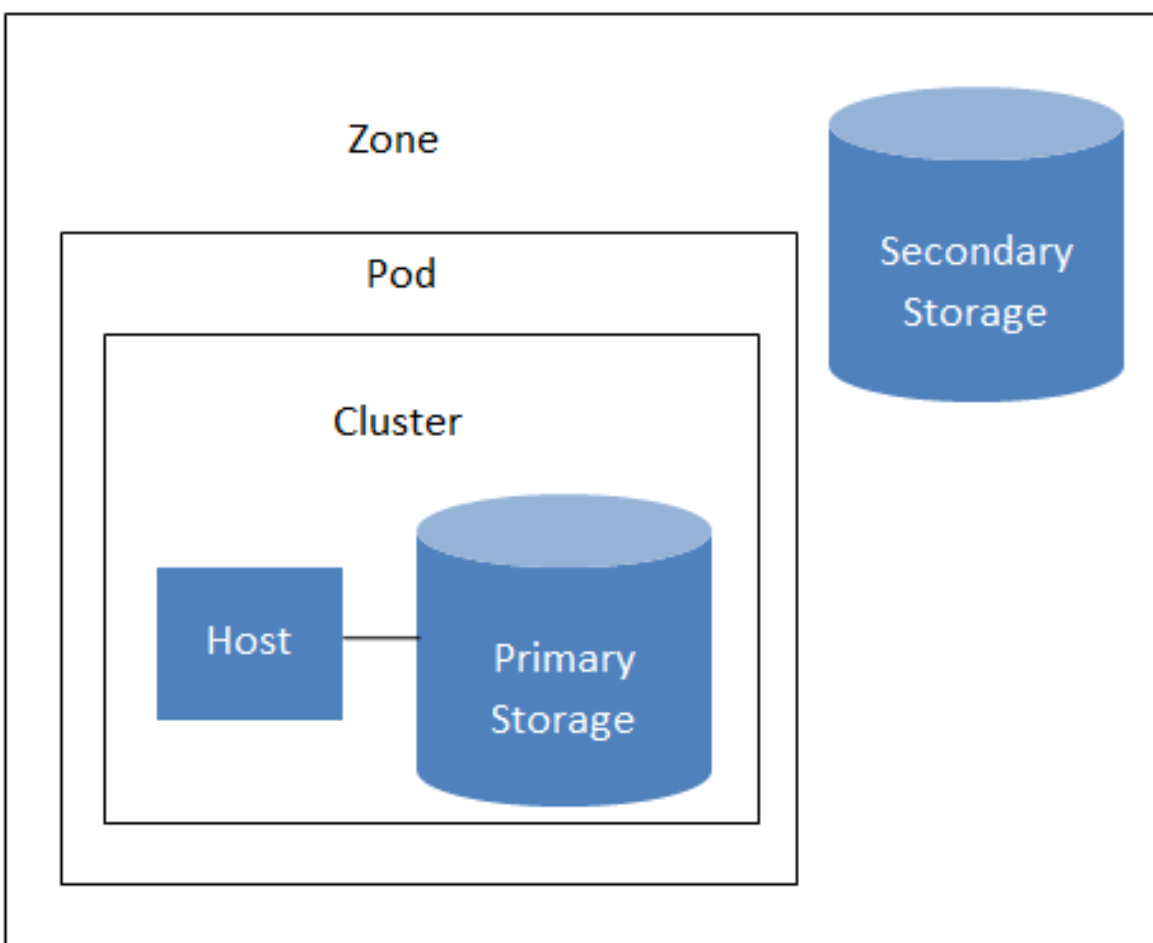
- Provides the web user interface for the administrator and a reference user interface for end users.
- Provides the APIs for CloudPlatform.
- Manages the assignment of guest VMs to particular hosts.
- Manages the assignment of public and private IP addresses to particular accounts.
- Manages the allocation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

3.3.2. 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:

- **Region:** To increase reliability of the cloud, you can optionally group resources into multiple geographic regions. A region consists of one or more zones.

- **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 can also be provisioned on a zone-wide basis. It stores the disk volumes for all the VMs running on hosts in that cluster.
- **Secondary storage** is associated with a zone, and it can also be provisioned as object storage that is available throughout the cloud. It stores templates, ISO images, and disk volume snapshots.



Nested organization of a zone

More Information

For more information, see Cloud Infrastructure Concepts in the Installation Guide.

3.3.3. Networking Overview

CloudPlatform offers two types of networking scenario:

- Basic. 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 and providing guest isolation.

For more details, see Network Setup in the Installation Guide.

What's New

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

4.1. What's New in the API for 4.2

The following describes any new major features of CloudPlatform 4.0 as it applies to API usage.

4.1.1. Added API Commands in 4.2

4.1.1.1. Secondary Storage

- `addImageStore` (Adds all types of secondary storage providers, S3/Swift/NFS)
- `createSecondaryStagingStore` (Adds a staging secondary storage in each zone)
- `listImageStores` (Lists all secondary storages, S3/Swift/NFS)
- `listSecondaryStagingStores` (Lists all staging secondary storages)
- `addS3` (Adds a Amazon Simple Storage Service instance.) It is recommended to use `addImageStore` instead.
- `listS3s` (Lists all the Amazon Simple Storage Service instances.) It is recommended to use `listImageStores` instead.

4.1.1.2. VM Snapshot

- `createVMSnapshot` (Creates a virtual machine snapshot; see Virtual Machine Snapshots for VMware in the CloudPlatform Administration Guide)
- `deleteVMSnapshot` (Deletes a virtual machine snapshot)
- `listVMSnapshot` (Shows a virtual machine snapshot)
- `revertToVMSnapshot` (Returns a virtual machine to the state and data saved in a given snapshot)

4.1.1.3. Load Balancer Health Check

- `createLBHealthCheckPolicy` (Creates a new health check policy for a load balancer rule; see Health Checks for Load Balancer Rules in the CloudPlatform Administration Guide)
- `deleteLBHealthCheckPolicy` (Deletes an existing health check policy from a load balancer rule)
- `listLBHealthCheckPolicies` (Displays the health check policy for a load balancer rule)

4.1.1.4. Egress Firewall Rules

- `createEgressFirewallRules` (Creates an egress firewall rule on the guest network; see Egress Firewall Rules in the CloudPlatform Administration Guide)
- `deleteEgressFirewallRules` (Deletes a egress firewall rule on the guest network.)
- `listEgressFirewallRules` (Lists the egress firewall rules configured for a guest network.)

4.1.1.5. SSH Key

- `resetSSHKeyForVirtualMachine` (Resets the SSHkey for virtual machine.)

4.1.1.6. Bare Metal

- `addBaremetalHost` (Adds a new host. Technically, this API command was present in v3.0.6, but its functionality was disabled. See Add a Bare Metal Host in the CloudPlatform Installation Guide)
- `addBaremetalDhcp` (Adds a DHCP server for bare metal hosts)
- `addBaremetalPxePingServer` (Adds a PXE PING server for bare metal hosts)
- `addBaremetalPxeKickStartServer` (Adds a PXE server for bare metal hosts)
- `listBaremetalDhcp` (Shows the DHCP servers currently defined for bare metal hosts)
- `listBaremetalPxePingServer` (Shows the PXE PING servers currently defined for bare metal hosts)

4.1.1.7. NIC

- `addNicToVirtualMachine` (Adds a new NIC to the specified VM on a selected network; see Configuring Multiple IP Addresses on a Single NIC in the CloudPlatform Administration Guide)
- `removeNicFromVirtualMachine` (Removes the specified NIC from a selected VM.)
- `updateDefaultNicForVirtualMachine` (Updates the specified NIC to be the default one for a selected VM.)
- `addIpToNic` (Assigns secondary IP to a NIC.)
- `removeIpFromNic` (Assigns secondary IP to a NIC.)
- `listNics` (Lists the NICs associated with a VM.)

4.1.1.8. Regions

- `addRegion` (Registers a Region into another Region; see Adding Regions in the CloudPlatform Administration Guide)
- `updateRegion` (Updates Region details: ID, Name, Endpoint, User API Key, and User Secret Key.)
- `removeRegion` (Removes a Region from current Region.)
- `listRegions` (Get all the Regions. They can be filtered by using the ID or Name.)

4.1.1.9. User

- `getUser` (This API can only be used by the Admin. Get user account details by using the API Key.)

4.1.1.10. API Throttling

- `getApiLimit` (Show number of remaining APIs for the invoking user in current window)
- `resetApiLimit` (For root admin, if `accountId` parameter is passed, it will reset count for that particular account, otherwise it will reset all counters)
- `resetApiLimit` (Reset the API count.)

4.1.1.11. Locking

- lockAccount (Locks an account)
- lockUser (Locks a user account)

4.1.1.12. VM Scaling

- scaleVirtualMachine (Scales the virtual machine to a new service offering.)

4.1.1.13. Migrate Volume

- migrateVirtualMachineWithVolume (Attempts migrating VM with its volumes to a different host.)
- listStorageProviders (Lists storage providers.)
- findStoragePoolsForMigration (Lists storage pools available for migrating a volume.)

4.1.1.14. Dedicated IP and VLAN

- dedicatePublicIpRange (Dedicates a Public IP range to an account.)
- releasePublicIpRange (Releases a Public IP range back to the system pool.)
- dedicateGuestVlanRange (Dedicates a guest VLAN range to an account.)
- releaseDedicatedGuestVlanRange (Releases a dedicated guest VLAN range to the system.)
- listDedicatedGuestVlanRanges (Lists dedicated guest VLAN ranges.)

4.1.1.15. Port Forwarding

- updatePortForwardingRule (Updates a port forwarding rule. Only the private port and the VM can be updated.)

4.1.1.16. Scale System VM

- scaleSystemVm (Scale the service offering for a systemVM, console proxy, or secondary storage.)

4.1.1.17. Deployment Planner

- listDeploymentPlanners (Lists all the deployment planners available.)

4.1.1.18. Archive and Delete Events and Alerts

- archiveEvents (Archive one or more events.)
- deleteEvents (Delete one or more events.)
- archiveAlerts (Archive one or more alerts.)
- deleteAlerts (Delete one or more alerts.)

4.1.1.19. Host Reservation

- releaseHostReservation (Releases host reservation.)

4.1.1.20. Resize Volume

- `resizeVolume` (Resizes a volume.)
- `updateVolume` (Updates the volume.)

4.1.1.21. Egress Firewall Rules

- `createEgressFirewallRule` (Creates a egress firewall rule for a given network.)
- `deleteEgressFirewallRule` (Deletes an egress firewall rule.)
- `listEgressFirewallRules` (Lists all egress firewall rules for network.)

4.1.1.22. Network ACL

- `updateNetworkACLItem` (Updates ACL item with specified ID.)
- `createNetworkACLList` (Creates a Network ACL for the given VPC.)
- `deleteNetworkACLList` (Deletes a Network ACL.)
- `replaceNetworkACLList` (Replaces ACL associated with a Network or private gateway.)
- `listNetworkACLLists` (Lists all network ACLs.)

4.1.1.23. Resource Detail

- `addResourceDetail` (Adds detail for the Resource.)
- `removeResourceDetail` (Removes detail for the Resource.)
- `listResourceDetails` (List resource details.)

4.1.1.24. Nicira Integration

- `addNiciraNvpDevice` (Adds a Nicira NVP device.)
- `deleteNiciraNvpDevice` (Deletes a Nicira NVP device.)
- `listNiciraNvpDevices` (Lists Nicira NVP devices.)
- `listNiciraNvpDeviceNetworks` (Lists network that are using a Nicira NVP device.)

4.1.1.25. BigSwitch VNS

- `addBigSwitchVnsDevice` (Adds a BigSwitch VNS device.)
- `deleteBigSwitchVnsDevice` (Deletes a BigSwitch VNS device.)
- `listBigSwitchVnsDevices` (Lists BigSwitch VNS devices.)

4.1.1.26. Simulator

- `configureSimulator` (Configures a simulator.)

4.1.1.27. API Discovery

- `listApis` (Lists all the available APIs on the server, provided by the API Discovery plugin.)

4.1.1.28. Global Load Balancer

- createGlobalLoadBalancerRule (Creates a global load balancer rule.)
- deleteGlobalLoadBalancerRule (Deletes a global load balancer rule.)
- updateGlobalLoadBalancerRule (update global load balancer rules.)
- listGlobalLoadBalancerRules (Lists load balancer rules.)
- assignToGlobalLoadBalancerRule (Assign load balancer rule or list of load balancer rules to a global load balancer rules.)
- removeFromGlobalLoadBalancerRule (Removes a load balancer rule association with global load balancer rule)

4.1.1.29. Load Balancer

- createLoadBalancer (Creates a Load Balancer)
- listLoadBalancers (Lists Load Balancers)
- deleteLoadBalancer (Deletes a load balancer)
- configureInternalLoadBalancerElement (Configures an Internal Load Balancer element.)
- createInternalLoadBalancerElement (Create an Internal Load Balancer element.)
- listInternalLoadBalancerElements (Lists all available Internal Load Balancer elements.)

4.1.1.30. Affinity Group

- createAffinityGroup (Creates an affinity or anti-affinity group.)
- deleteAffinityGroup (Deletes an affinity group.)
- listAffinityGroups (Lists all the affinity groups.)
- updateVMAffinityGroup (Updates the affinity or anti-affinity group associations of a VM. The VM has to be stopped and restarted for the new properties to take effect.)
- listAffinityGroupTypes (Lists affinity group types available.)

4.1.1.31. Portable IP

- createPortableIpRange (Adds a range of portable IP ranges to a Region.)
- deletePortableIpRange (Deletes a range of portable IP ranges associated with a Region.)
- listPortableIpRanges (Lists portable IP ranges.)

4.1.1.32. Internal Load Balancer VM

- stopInternalLoadBalancerVM (Stops an Internal LB VM.)
- startInternalLoadBalancerVM (Starts an existing Internal LB VM.)
- listInternalLoadBalancerVMs (List internal LB VMs.)

4.1.1.33. Network Isolation

- `listNetworkIsolationMethods` (Lists supported methods of network isolation.)

4.1.1.34. Dedicated Resources

- `dedicateZone` (Dedicates a zone.)
- `dedicatePod` (Dedicates a pod.)
- `dedicateCluster` (Dedicate an existing cluster.)
- `dedicateHost` (Dedicates a host.)
- `releaseDedicatedZone` (Release dedication of zone.)
- `releaseDedicatedPod` (Release dedication for the pod.)
- `releaseDedicatedCluster` (Release dedication for cluster.)
- `releaseDedicatedHost` (Release dedication for host.)
- `listDedicatedZones` (List dedicated zones.)
- `listDedicatedPods` (Lists dedicated pods.)
- `listDedicatedClusters` (Lists dedicated clusters.)
- `listDedicatedHosts` (Lists dedicated hosts.)

4.1.2. Changed API Commands in 4.2

API Commands	Description
<code>listNetworkACLs</code>	The following new request parameters are added: <code>aclid</code> (optional), <code>action</code> (optional), <code>protocol</code> (optional) The following new response parameters are added: <code>aclid</code> , <code>action</code> , <code>number</code>
<code>copyTemplate</code>	The following new response parameters are added: <code>isdynamicallyscalable</code> , <code>sshkeyenabled</code>
<code>listRouters</code>	The following new response parameters are added: <code>ip6dns1</code> , <code>ip6dns2</code> , <code>role</code>
<code>updateConfiguration</code>	The following new request parameters are added: <code>accountid</code> (optional), <code>clusterid</code> (optional), <code>storageid</code> (optional), <code>zoneid</code> (optional) The following new response parameters are added: <code>id</code> , <code>scope</code>
<code>listVolumes</code>	The following request parameter is removed: <code>details</code> The following new response parameter is added: <code>displayvolume</code>

API Commands	Description
suspendProject	The following new response parameters are added: cpuavailable, cpulimit, cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpcotal
listRemoteAccessVpns	The following new response parameters are added: id
registerTemplate	The following new request parameters are added: imagestoreuuid (optional), isdynamicallyscalable (optional), isrouting (optional) The following new response parameters are added: isdynamicallyscalable, sshkeyenabled
addTrafficMonitor	The following response parameters are removed: privateinterface, privatezone, publicinterface, publiczone, usageinterface, username
createTemplate	The following response parameters are removed: clusterid, clustername, disksizeallocated, disksizetotal, disksizeused, ipaddress, path, podid, podname, state, tags, type The following new response parameters are added: account, accountid, bootable, checksum, crossZones, details, displaytext, domain, domainid, format, hostid, hostname, hypervisor, isdynamicallyscalable, isextractable, isfeatured, ispublic, isready, ostypeid, ostypeid, passwordenabled, project, projectid, removed, size, sourcetemplateid, sshkeyenabled, status, templatetag, templatetype, tags
listLoadBalancerRuleInstances	The following new response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
migrateVolume	The following new request parameters is added: livemigrate (optional) The following new response parameters is added: displayvolume
createAccount	The following new request parameters are added: accountid (optional), userid (optional)

API Commands	Description
	<p>The following new response parameters are added: accountdetails, cpuavailable, cpulimit, cputotal, defaultzoneid, ipavailable, iplimit, iptotal, iscleanuprequired, isdefault, memoryavailable, memorylimit, memorytotal, name, networkavailable, networkdomain, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, projectavailable, projectlimit, projecttotal, receivedbytes, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, sentbytes, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpctotal, user</p> <p>The following parameters are removed: account, accountid, apikey, created, email, firstname, lastname, secretkey, timezone, username</p>
updatePhysicalNetwork	The following new request parameters is added: removevlan (optional)
listTrafficMonitors	The following response parameters are removed: privateinterface, privatezone, publicinterface, publiczone, usageinterface, username
attachIso	The following new response parameters are added: diskiread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
listProjects	The following new request parameters are added: cpuavailable, cpulimit, cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpctotal
enableAccount	The following new response parameters are added: cpuavailable, cpulimit, cputotal, isdefault, memoryavailable, memorylimit, memorytotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal

API Commands	Description
listPublicIpsAddresses	The following new response parameters are added: isportable, vmipaddress
enableStorageMaintenance	The following new response parameters are added: hypervisor, scope, suitableformigration
listLoadBalancerRules	The following new request parameters is added: networkid (optional) The following new response parameters is added: networkid
stopRouter	The following new response parameters are added: ip6dns1, ip6dns2, role
listClusters	The following new response parameters are added: cpuovercommitratio, memoryovercommitratio
attachVolume	The following new response parameter is added: displayvolume
updateVPCOffering	The following request parameters is made mandatory: id
resetSSHKeyForVirtualMachine	The following new request parameter is added: keypair (required) The following parameter is removed: name The following new response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
updateCluster	The following new request parameters are added: cpuovercommitratio (optional), memoryovercommitratio (optional) The following new response parameters are added: cpuovercommitratio, memoryovercommitratio
listPrivateGateways	The following new response parameters are added: aclid, sourcenatsupported
ldapConfig	The following new request parameters are added: listall (optional) The following parameters has been made optional: searchbase, hostname, queryfilter The following new response parameter is added: ssl
listTemplates	The following new response parameters are added: isdynamicallyscalable, sshkeyenabled
listNetworks	The following new response parameters are added: aclid, displaynetwork, ip6cidr,

API Commands	Description
	ip6gateway, ispersistent, networkcidr, reservediprange
restartNetwork	The following new response parameters are added: isportable, vmipaddress
prepareTemplate	The following new response parameters are added: isdynamicallyscalable, sshkeyenabled
rebootVirtualMachine	The following new response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
changeServiceForRouter	The following new request parameters are added: aclid (optional), action (optional), protocol (optional) The following new response parameters are added: id, scope
updateZone	The following new request parameters are added: ip6dns1 (optional), ip6dns2 (optional) The following new response parameters are added: ip6dns1, ip6dns2
ldapRemove	The following new response parameters are added: ssl
updateServiceOffering	The following new response parameters are added: deploymentplanner, isvolatile
updateStoragePool	The following new response parameters are added: hypervisor, scope, suitableformigration
listFirewallRules	The following request parameter is removed: traffictype The following new response parameters are added: networkid
updateUser	The following new response parameters are added: iscallerchilddomain, isdefault
updateProject	The following new response parameters are added: cpuavailable, cpulimit, cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpcplimit, vpcptotal

API Commands	Description
updateTemplate	<p>The following new request parameters are added: isdynamicallyscalable (optional), isrouting (optional)</p> <p>The following new response parameters are added: isdynamicallyscalable, sshkeyenabled</p>
disableUser	<p>The following new response parameters are added: iscallerchilddomain, isdefault</p>
activateProject	<p>The following new response parameters are added: cpuavailable, cpulimit, cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpcotal</p>
createNetworkACL	<p>The following new request parameters are added: aclid (optional), action (optional), number (optional)</p> <p>The following request parameter is now optional: networkid</p> <p>The following new response parameters are added: aclid, action, number</p>
enableStaticNat	<p>The following new request parameters are added: vmguestip (optional)</p>
registerIso	<p>The following new request parameters are added: imagestoreuuid (optional), isdynamicallyscalable (optional)</p> <p>The following new response parameters are added: isdynamicallyscalable, sshkeyenabled</p>
createIpForwardingRule	<p>The following new response parameter is added: vmguestip</p>
resetPasswordForVirtualMachine	<p>The following new response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup</p>
createVolume	<p>The following new request parameter is added: displayvolume (optional)</p> <p>The following new response parameter is added: displayvolume</p>

API Commands	Description
startRouter	The following new response parameters are added: ip6dns1, ip6dns2, role
listCapabilities	The following new response parameters are added: apilimitinterval and apilimitmax.
createServiceOffering	<p>The following new request parameters are added: deploymentplanner (optional), isvolatile (optional), serviceofferingdetails (optional).</p> <p>isvolatile indicates whether the service offering includes Volatile VM capability, which will discard the VM's root disk and create a new one on reboot.</p> <p>The following new response parameters are added: deploymentplanner, isvolatile</p>
restoreVirtualMachine	<p>The following request parameter is added: templateID (optional). This is used to point to the new template ID when the base image is updated.</p> <p>The following response parameters are added: diskiread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup</p>
createNetwork	<p>The following new request parameters are added: aclid (optional), displaynetwork (optional), endipv6 (optional), ip6cidr (optional), ip6gateway (optional), isolatedpvlan (optional), startipv6 (optional)</p> <p>The following new response parameters are added: aclid, displaynetwork, ip6cidr, ip6gateway, ispersistent, networkcidr, reservediprange</p>
createVlanIpRange	<p>The following new request parameters are added: startipv6, endipv6, ip6gateway, ip6cidr</p> <p>Changed parameters: startip (is now optional)</p> <p>The following new response parameters are added: startipv6, endipv6, ip6gateway, ip6cidr</p>
CreateZone	<p>The following new request parameters are added: ip6dns1, ip6dns2</p> <p>The following new response parameters are added: ip6dns1, ip6dns2</p>
deployVirtualMachine	The following request parameters are added: affinitygroupids (optional), affinitygroupnames (optional), displayvm (optional), ip6address (optional)

API Commands	Description
	<p>The following request parameter is modified: iptonetworklist has a new possible value, ipv6</p> <p>The following new response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup</p>
createNetworkOffering	<p>The following request parameters are added: details (optional), egressdefaultpolicy (optional), ispersistent (optional)</p> <p>ispersistent determines if the network or network offering created or listed by using this offering are persistent or not.</p> <p>The following response parameters are added: details, egressdefaultpolicy, ispersistent</p>
listNetworks	<p>The following request parameters is added: isPersistent.</p> <p>This parameter determines if the network or network offering created or listed by using this offering are persistent or not.</p>
listNetworkOfferings	<p>The following request parameters is added: isPersistent.</p> <p>This parameter determines if the network or network offering created or listed by using this offering are persistent or not.</p> <p>For listNetworkOfferings, the following response parameter has been added: details, egressdefaultpolicy, ispersistent</p>
addF5LoadBalancer configureNetscalerLoadBalancer addNetscalerLoadBalancer listF5LoadBalancers configureF5LoadBalancer listNetscalerLoadBalancers	<p>The following response parameter is removed: inline.</p>
listRouters	<p>For nic responses, the following fields have been added.</p> <ul style="list-style-type: none"> • ip6address • ip6gateway • ip6cidr

API Commands	Description
listVirtualMachines	<p>The following request parameters are added: affinitygroupid (optional), vpcid (optional)</p> <p>The following response parameters are added: diskiread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup</p>
listRouters listZones	<p>For DomainRouter and DataCenter response, the following fields have been added.</p> <ul style="list-style-type: none"> • ip6dns1 • ip6dns2 <p>For listZones, the following optional request parameters are added: name, networktype</p>
listFirewallRules createFirewallRule	<p>The following request parameter is added: traffictype (optional).</p> <p>The following response parameter is added: networkid</p>
listUsageRecords	<p>The following response parameter is added: virtualsize.</p>
deletelso	<p>The following request parameter is removed: forced</p>
addCluster	<p>The following request parameters are added: cpuovercommitratio (optional), guestvswitchtype (optional), guestvswitchtype (optional), memoryovercommitratio (optional), publicvswitchtype (optional), publicvswitchtype (optional)</p> <p>The following request parameters are added: cpuovercommitratio, memoryovercommitratio</p>
updateCluster	<p>The following request parameters are added: cpuovercommitratio, ramovercommitratio</p>
createStoragePool	<p>The following request parameters are added: hypervisor (optional), provider (optional), scope (optional)</p> <p>The following request parameters have been made mandatory: podid, clusterid</p> <p>The following response parameter has been added: hypervisor, scope, suitableformigration</p>
listStoragePools	<p>The following request parameter is added: scope (optional)</p> <p>The following response parameters are added: hypervisor, scope, suitableformigration</p>

API Commands	Description
updateDiskOffering	The following response parameter is added: displayoffering
changeServiceForVirtualMachine	The following response parameter are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
recoverVirtualMachine	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
listCapabilities	The following response parameters are added: apilimitinterval, apilimitmax
createRemoteAccessVpn	The following response parameters are added: id
startVirtualMachine	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
detachIso	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
updateVPC	The following request parameters has been made mandatory: id, name
associateIpAddress	The following request parameters are added: isportable (optional), regionid (optional) The following response parameters are added: isportable, vmipaddress
listProjectAccounts	The following response parameters are added: cpuavailable, cpulimit, cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpcctotal
disableAccount	The following response parameters are added: cpuavailable, cpulimit, cputotal, isdefault, memoryavailable, memorylimit, memorytotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal

API Commands	Description
listPortForwardingRules	The following response parameters are added: vmguestip
migrateVirtualMachine	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
cancelStorageMaintenance	The following response parameters are added: hypervisor, scope, suitableformigration
createPortForwardingRule	The following request parameter is added: vmguestip (optional) The following response parameter is added: vmguestip
addVpnUser	The following response parameter is added: state
createVPCOffering	The following request parameter is added: serviceproviderlist (optional)
assignVirtualMachine	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
listConditions	The following response parameters are added: account, counter, domain, domainid, project, projectid, relationaloperator, threshold Removed response parameters: name, source, value
createPrivateGateway	The following request parameters are added: aclid (optional), sourcenatsupported (optional) The following response parameters are added: aclid, sourcenatsupported
updateVirtualMachine	The following request parameters are added: displayvm (optional), isdynamicallyscalable (optional) The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
destroyRouter	The following response parameters are added: ip6dns1, ip6dns2, role
listServiceOfferings	The following response parameters are added: deploymentplanner, isvolatile
listUsageRecords	The following response parameters are removed: virtualsize
createProject	The following response parameters are added: cpuavailable, cpulimit,

API Commands	Description
	<p>cputotal, ipavailable, iplimit, iptotal, memoryavailable, memorylimit, memorytotal, networkavailable, networklimit, networktotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal, snapshotavailable, snapshotlimit, snapshottotal, templateavailable, templatelimit, templatetotal, vmavailable, vmlimit, vmrunning, vmstopped, vmtotal, volumeavailable, volumelimit, volumetotal, vpcavailable, vpclimit, vpctotal</p>
enableUser	<p>The following response parameters are added: iscallerchilddomain, isdefault</p>
createLoadBalancerRule	<p>The following response parameter is added: networkid</p>
updateAccount	<p>The following response parameters are added: cpuavailable, cpulimit, cputotal, isdefault, memoryavailable, memorylimit, memorytotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal</p>
copyIso	<p>The following response parameters are added: isdynamicallyscalable, sshkeyenabled</p>
uploadVolume	<p>The following request parameters are added: imagestoreuuid (optional), projectid (optional)</p> <p>The following response parameters are added: displayvolume</p>
createDomain	<p>The following request parameter is added: domainid (optional)</p>
stopVirtualMachine	<p>The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup</p>
listAccounts	<p>The following response parameters are added: cpuavailable, cpulimit, cputotal, isdefault, memoryavailable, memorylimit, memorytotal, primarystorageavailable, primarystoragelimit, primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal</p>
createSnapshot	<p>The following response parameter is added: zoneid</p>
updateIso	<p>The following request parameters are added: isdynamicallyscalable (optional), isrouting (optional)</p> <p>The following response parameters are added: isdynamicallyscalable, sshkeyenabled</p>

API Commands	Description
listIpForwardingRules	The following response parameter is added: vmguestip
updateNetwork	<p>The following request parameters are added: displaynetwork (optional), guestvmcidr (optional)</p> <p>The following response parameters are added: aclid, displaynetwork, ip6cidr, ip6gateway, ispersistent, networkcidr, reservediprange</p>
destroyVirtualMachine	The following response parameters are added: diskioread, diskiowrite, diskkbsread, diskkbswrite, displayvm, isdynamicallyscalable, affinitygroup
createDiskOffering	<p>The following request parameter is added: displayoffering (optional)</p> <p>The following response parameter is added: displayoffering</p>
rebootRouter	The following response parameters are added: ip6dns1, ip6dns2, role
listConfigurations	<p>The following request parameters are added: accountid (optional), clusterid (optional), storageid (optional), zoneid (optional)</p> <p>The following response parameters are added: id, scope</p>
createUser	<p>The following request parameter is added: userid (optional)</p> <p>The following response parameters are added: iscallerchilddomain, isdefault</p>
listDiskOfferings	The following response parameter is added: displayoffering
detachVolume	The following response parameter is added: displayvolume
deleteUser	<p>The following response parameters are added: displaytext, success</p> <p>Removed parameters: id, account, accountid, accounttype, apikey, created, domain, domainid, email, firstname, lastname, secretkey, state, timezone, username</p>
listSnapshots	<p>The following request parameter is added: zoneid (optional)</p> <p>The following response parameter is added: zoneid</p>
markDefaultZoneForAccount	The following response parameters are added: cpuavailable, cpulimit, cputotal, isdefault, memoryavailable, memorylimit, memorytotal, primarystorageavailable, primarystoragelimit,

API Commands	Description
	primarystoragetotal, secondarystorageavailable, secondarystoragelimit, secondarystoragetotal
restartVPC	The following request parameters are made mandatory: id
updateHypervisorCapabilities	The following response parameters are added: hypervisor, hypervisorversion, maxdatavolumeslimit, maxguestslimit, maxhostspercluster, securitygroupenabled, storagemotionenabled Removed parameters: cpunumber, cpuspeed, created, defaultuse, displaytext, domain, domainid, hosttags, issystem, limitcpuuse, memory, name, networkrate, offerha, storagetype, systemvmtype, tags
updateLoadBalancerRule	The following response parameter is added: networkid
listVlanIpRanges	The following response parameters are added: endipv6, ip6cidr, ip6gateway, startipv6
listHypervisorCapabilities	The following response parameters are added: maxdatavolumeslimit, maxhostspercluster, storagemotionenabled
updateNetworkOffering	The following response parameters are added: details, egressdefaultpolicy, ispersistent
createVirtualRouterElement	The following request parameters are added: providertype (optional)
listVpnUsers	The following response parameter is added: state
listUsers	The following response parameters are added: iscallerchilddomain, isdefault
listSupportedNetworkServices	The following response parameter is added: provider
listIsos	The following response parameters are added: isdynamicallyscalable, sshkeyenabled

4.1.3. Deprecated APIs

- addExternalLoadBalancer (Adds F5 external load balancer appliance.)
- deleteExternalLoadBalancer (Deletes a F5 external load balancer appliance added in a zone.)
- listExternalLoadBalancers (Lists F5 external load balancer appliances added in a zone.)

4.2. What's New in the API for 3.0.x?

The following describes any new major features of CloudPlatform 3.0.x as it applies to API usage.

4.2.1. IPv6 Support in CloudPlatform

CloudPlatform 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, 5001:0dt8:83a3:1012:1000:8s2e:0870:7454. CloudPlatform 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 [Section 4.2.9, "Changed API Commands"](#).

4.2.1.1. 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](#)¹ 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.



Note

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://nfs1.lab.vmops.com/templates/routing/debian/ipv6/>.

¹ <http://tools.ietf.org/html/rfc3315>

- The concept of Default Network applies to IPv6 networks. However, unlike IPv4 CloudPlatform 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 CloudPlatform 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.

4.2.1.2. Limitations of IPv6 in CloudPlatform

The following are not yet supported:

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

4.2.1.3. 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`.



Note

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
```

4.2.2. Additional VMX Settings

A VMX (.vmx) file is the primary configuration file for a virtual machine. When a new VM is created, information on the operating system, disk sizes, and networking is stored in this file. The VM actively writes to its .vmx file for all the configuration changes. The VMX file is typically located in the directory where the VM is created. In Windows Vista / Windows 7 / Windows Server 2008, the default location is `C:\Users\<your_user_name>\My Documents\Virtual Machines\<virtual_machine_name>.vmx`. In Linux, `vmware-cmd -l` lists the full path to all the registered VMX files. Any manual additions to the .vmx file from ESX/ESXi are overwritten by the entries stored in the vCenter Server database. Therefore, before you edit a .vmx file, first remove the VM from the vCenter server's inventory and register the VM again after editing.

The CloudStack API that supports passing some of the VMX settings is `registerTemplate`. The supported parameters are `rootDiskController`, `nicAdapter`, and `keyboard`. In addition to these existing VMX parameters, you can now use the `keyboard.typematicMinDelay` parameter in the `registerTemplate` API call. This parameter controls the amount of delay for the repeated key strokes on remote consoles. For more information on `keyboard.typematicMinDelay`, see [keyboard.typematicMinDelay²](#).

4.2.3. Resetting SSH Keys to Access VMs

With the new API command `resetSSHKeyForVirtualMachine`, a user can set or reset the SSH keypair assigned to a virtual machine. With the addition of this feature, a lost or compromised SSH keypair can be changed, and the user can access the VM by using the new keypair. Just create or register a new keypair, then call `resetSSHKeyForVirtualMachine`.

4.2.4. XenServer Hotfix Notification

You can get notifications about new Citrix XenServer software updates as they become available. Use the following new API commands to get the hotfix notification:

- **listHostUpdates**—Specifies which patches have and have not been applied to a given host.
- **listHostsWithPendingUpdates**—Specifies which hosts have not received a given patch.

In this way, administrators can more easily be aware and take action when a hotfix is required to keep the XenServer hypervisor hosts up to date. CloudPlatform will give notification only for hotfixes that are applicable to the XenServer software version currently running on hosts within the cloud. Before generating the alert, CloudPlatform will check to be sure the hotfix has not already been installed.

CloudPlatform checks for new hotfixes once a week. It can be configured to check for hotfixes more or less often through the global setting `update.check.interval`.

4.2.5. Enabling Port 8096

Port 8096, which allows API calls without authentication, is closed and disabled by default on any fresh installations of CloudPlatform 3.0.1 or greater. 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.

4.2.6. Stopped VM

CloudPlatform 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.

² http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=196

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.

4.2.7. 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.

listTemplates Flag	Description
community	Returns templates that have been marked as public but not featured.
all	Returns all templates (only usable by admins).

The CloudPlatform 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.

4.2.8. Added API commands

4.2.8.1. Added in 3.0.6

- createEgressFirewallRules (creates an egress firewall rule on the guest network.)
- deleteEgressFirewallRules (deletes a egress firewall rule on the guest network.)
- listEgressFirewallRules (lists the egress firewall rules configured for a guest network.)
- resetSSHKeyForVirtualMachine (Resets the SSHkey for virtual machine.)
- addBaremetalHost (Adds a new host.)

4.2.8.2. Added in 3.0.5

- 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)
- createCounter (Adds a metric counter)
- createCondition (Creates a condition)

- createAutoScalePolicy (Creates an AutoScale policy for a provision or deprovision action.)
- createAutoScaleVmProfile (Creates a profile that contains information about the virtual machine which will be provisioned automatically by AutoScale feature)
- createAutoScaleVmGroup (Creates and automatically starts a virtual machine based on a service offering, disk offering, and template)
- deleteCounter (Deletes a counter)
- deleteCondition (Deletes a condition)
- deleteAutoScalePolicy (Deletes an AutoScale policy)
- deleteAutoScaleVmProfile (Deletes a AutoScale vm profile)
- deleteAutoScaleVmGroup (Deletes a AutoScale vm group)
- listCounters (List the counters)
- listConditions (List Conditions for the specific user)
- listAutoScalePolicies (Lists AutoScale policies)
- listAutoScaleVmProfiles (Lists AutoScale vm profiles)
- listAutoScaleVmGroups (Lists AutoScale vm groups)
- enableAutoScaleVmGroup (Enables an AutoScale vm group)
- disableAutoScaleVmGroup (Disables an AutoScale vm group)
- updateAutoScalePolicy (Updates an existing AutoScale policy)
- updateAutoScaleVmProfile (Updates an existing AutoScale vm profile)
- updateAutoScaleVmGroup (Updates an existing AutoScale vm group)

4.2.8.3. Added in 3.0.3

- enableCiscoNexusVSM (Enables Nexus 1000v dvSwitch in CloudPlatform.)
- disableCiscoNexusVSM (Disables Nexus 1000v dvSwitch in CloudPlatform.)
- deleteCiscoNexusVSM (Deletes Nexus 1000v dvSwitch in CloudPlatform.)
- listCiscoNexusVSMs (Lists the control VLAN ID, packet VLAN ID, and data VLAN ID, as well as the IP address of the Nexus 1000v dvSwitch.)

4.2.8.4. 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.

4.2.8.5. 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.

4.2.8.6. 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 (Update a 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 CloudPlatform 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.)		

4.2.9. Changed API Commands

4.2.9.1. Changed API Commands in 3.0.7

API Commands	Description
createNetworkOffering	<p>The following request parameters have been added:</p> <ul style="list-style-type: none"> isPersistent startipv6 endipv6 ip6gateway ip6cidr
listNetworkOfferings listNetworks	<p>The following request parameters have been added:</p> <ul style="list-style-type: none"> isPersistent <p>This parameter determines if the network or network offering listed are persistent or not.</p>


API Commands	Description
	<ul style="list-style-type: none"> ip6gateway ip6cidr
createVlanIpRange	<p>The following request parameters have been added:</p> <ul style="list-style-type: none"> startipv6 endipv6 ip6gateway ip6cidr
deployVirtualMachine	<p>The following parameter has been added: ip6Address.</p> <p>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"> startipv6 endipv6 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

4.2.9.2. Changed API Commands in 3.0.6

API Commands	Description
addF5LoadBalancer configureNetscalerLoadBalancer addNetscalerLoadBalancer	<p>The following response parameter is removed: inline.</p>

API Commands	Description
listF5LoadBalancers configureF5LoadBalancer listNetscalerLoadBalancers	
listFirewallRules createFirewallRule	The following request parameter is added: traffictype (optional).
listUsageRecords	The following response parameter is added: virtualsize.
deleteIso	The following request parameter is added: forced (optional).
createStoragePool	The following request parameters are made mandatory: <ul style="list-style-type: none"> • podid • clusterid
listZones	The following request parameters is added: securitygroupenabled

4.2.9.3. Changed API Commands in 3.0.5

API Commands	Description
copyTemplate prepareTemplate registerTemplate updateTemplate createProject activateProject suspendProject updateProject listProjectAccounts createVolume migrateVolume attachVolume detachVolume uploadVolume createSecurityGroup	<p>The commands in this list have a single new response parameter, and no other changes.</p> <p>New response parameter: tags(*)</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <p>Note</p> <p>Many other commands also have the new tags(*) parameter in addition to other changes; those commands are listed separately.</p> </div>

Chapter 4. What's New

API Commands	Description
registerIso copyIso updateIso createIpForwardingRule listIpForwardingRules createLoadBalancerRule updateLoadBalancerRule createSnapshot	
rebootVirtualMachine attachIso detachIso listLoadBalancerRuleInstances resetPasswordForVirtualMachine changeServiceForVirtualMachine recoverVirtualMachine startVirtualMachine migrateVirtualMachine deployVirtualMachine assignVirtualMachine updateVirtualMachine restoreVirtualMachine stopVirtualMachine destroyVirtualMachine	<p>The commands in this list have two new response parameters, and no other changes.</p> <p>New response parameters: keypair, tags(*)</p>
listSecurityGroups listFirewallRules listPortForwardingRules listSnapshots listIcos listProjects listTemplates	<p>The commands in this list have the following new parameters, and no other changes.</p> <p>New request parameter: tags (optional)</p> <p>New response parameter: tags(*)</p>

API Commands	Description
listLoadBalancerRules	
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
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, vpclimit, vpcctotal
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(*)
listPublicIpAddresses	New request parameters: tags (optional), vpcid (optional) New response parameters: vpcid, tags(*)
listNetworks	New request parameters: canusefordeploy (optional), forvpc (optional), tags (optional), vpcid (optional)

API Commands	Description
	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) 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

4.2.10. Removed API commands

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

4.2.11. Added CloudPlatform Error Codes

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

- 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 : "com.cloud.api.ServerApiException"

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=miVr6X7u6lN_sdahOBpjNejPgEst35eXqjB8CG20YI3yaxXcgpyuaIRmFI_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=miVr6X7u6lN_sdahOBpjNejPgEst35eXqjB8CG20YI3yaxXcgpyuaIRmFI_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](#).

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

5.1. 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.

Chapter 5. Making API Requests

```
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_sdah
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_sdahobpjnejpggest35exq-
jb8cg20yi3yaxxcgpyuaairmfi_ejtvwz0nukkjbpmY3y2bciKwFq&command=deployvirtualmachine&diskofferingid=1&serviceofferingid=1
```

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=miVr6
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU
%2FcaiK8RAP001hU%3D
```

5.2. 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_sdahOBpjNejPgEs
jB8CG20YI3yaxXcgyuaIRmFI_EJTVwZ0nUkkJbPmY3y2bciKwFQ&signature=Lxx1DM40AjcXU%2FcaiK8RAP001hU
%3D
```

5.3. 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.

5.3.1. 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.

5.3.2. Limitations on API Throttling

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



Note

Even with these limitations, CloudPlatform 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, CloudPlatform 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.

Responses

6.1. 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"
      }
    ]
  }
}
```

6.2. 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.

6.3. 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.

Asynchronous Commands

CloudPlatform API 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.

7.1. 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.

7.2. 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>
```

Chapter 7. Asynchronous Commands

```
<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>
```


Working With Usage Data

The Usage Server provides aggregated usage records which you can use to create billing integration for CloudPlatform. 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 CloudPlatform Administration Guide.

8.1. Usage Record Format

8.1.1. 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

8.1.2. 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

8.1.3. 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.

8.1.4. 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

8.1.5. 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

8.1.6. 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

8.1.7. 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

8.1.8. 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

8.2. 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.

8.3. Example response from listUsageRecords

All CloudPlatform 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>
```

8.4. 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.

Third-Party UI Plugin Framework

Using the new third-party plugin framework, you can write and install extensions to CloudPlatform. The installed and enabled plugins will appear in the UI alongside the other features. The code for the plugin is simply placed in a special directory within CloudPlatform's installed code at any time after CloudPlatform installation. The new plugin appears only when it is enabled by the cloud administrator.



The left navigation bar of the CloudPlatform UI has a new Plugins button to help you work with UI plugins.

9.1. How to Write a Plugin: Overview

The basic procedure for writing a plugin is:

1. Write the code and create the other files needed. You will need the plugin code itself (in Javascript), a thumbnail image, the plugin listing, and a CSS file.



File Hierarchy

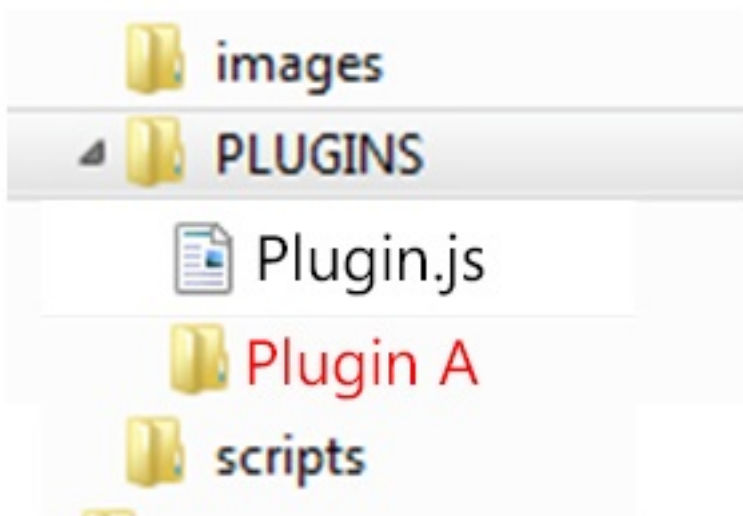
- plugin
- thumbnail
- description
- etc...

All UI plugins have the following set of files:

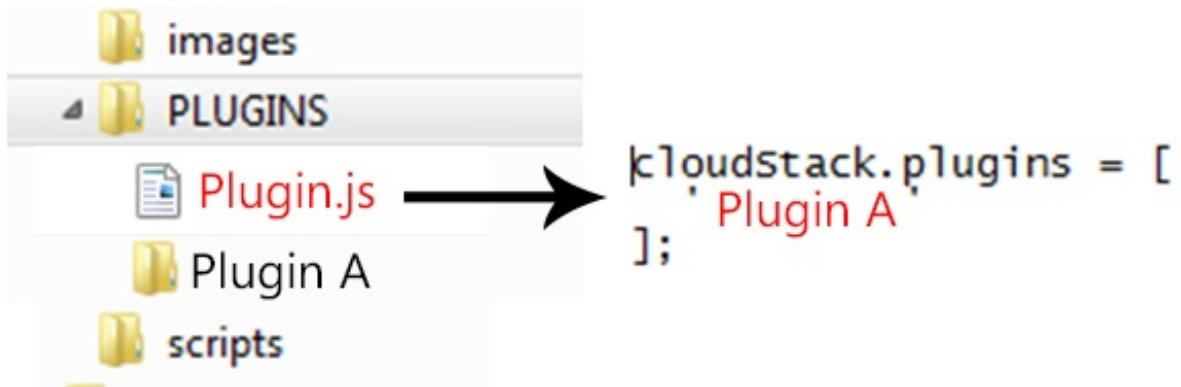
Chapter 9. Third-Party UI Plugin Framework

```
+-- cloudstack/  
  +-- ui/  
    +-- plugins/  
      +-- csMyFirstPlugin/  
        +-- config.js      --> Plugin metadata (title, author, vendor URL, etc.)  
        +-- icon.png      --> Icon, shown on side nav bar and plugin listing  
                               (should be square, and ~50x50px)  
        +-- csMyFirstPlugin.css --> CSS file, loaded automatically when plugin loads  
        +-- csMyFirstPlugin.js --> Main JS file, containing plugin code
```

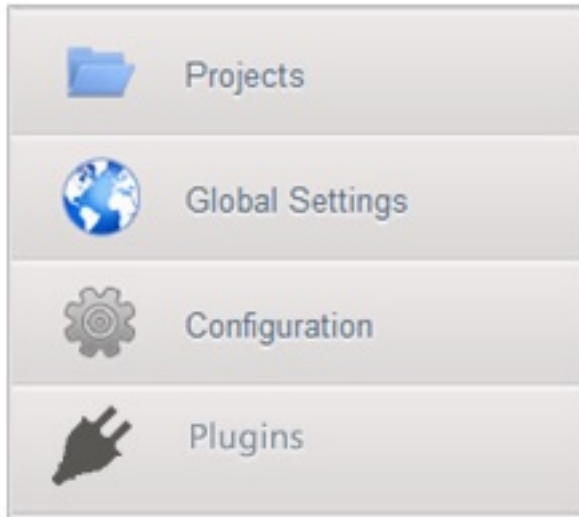
2. The CloudPlatform administrator adds the folder containing your plugin code under the CloudPlatform PLUGINS folder.



3. The administrator also adds the name of your plugin to the plugin.js file in the PLUGINS folder.



4. The next time the user refreshes the UI in the browser, your plugin will appear in the left navigation bar.



9.2. How to Write a Plugin: Implementation Details

This section requires an understanding of JavaScript and the CloudPlatform API. You don't need knowledge of specific frameworks for this tutorial (jQuery, etc.), since the CloudPlatform UI handles the front-end rendering for you.

There is much more to the CloudPlatform UI framework than can be described here. The UI is very flexible to handle many use cases, so there are countless options and variations. The best reference right now is to read the existing code for the main UI, which is in the /ui folder. Plugins are written in a very similar way to the main UI.

1. Create the directory to hold your plugin.

All plugins are composed of set of required files in the directory /ui/plugins/pluginID, where pluginID is a short name for your plugin. It's recommended that you prefix your folder name (for example, bfMyPlugin) to avoid naming conflicts with other people's plugins.

In this example, the plugin is named csMyFirstPlugin.

```
$ cd cloudstack/ui/plugins
$ mkdir csMyFirstPlugin
$ ls -l

total 8
drwxr-xr-x  2 bgregory  staff   68 Feb 11 14:44 csMyFirstPlugin
-rw-r--r--  1 bgregory  staff  101 Feb 11 14:26 plugins.js
```

2. Change to your new plugin directory.

```
$ cd csMyFirstPlugin
```

3. Set up the listing.

Add the file config.js, using your favorite editor.

```
$ vi config.js
```

Add the following content to `config.js`. This information will be displayed on the plugin listing page in the UI:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin.config = {
    title: 'My first plugin',
    desc: 'Tutorial plugin',
    externalLink: 'http://www.cloudstack.org/',
    authorName: 'Test Plugin Developer',
    authorEmail: 'plugin.developer@example.com'
  };
}(cloudStack));
```

4. Add a new main section.

Add the file `csMyFirstPlugin.js`, using your favorite editor.

```
$ vi csMyFirstPlugin.js
```

Add the following content to `csMyFirstPlugin.js`:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin = function(plugin) {
    plugin.ui.addSection({
      id: 'csMyFirstPlugin',
      title: 'My Plugin',
      preFilter: function(args) {
        return isAdmin();
      },
      show: function() {
        return $('<div>').html('Content will go here');
      }
    });
  };
}(cloudStack));
```

5. Register the plugin.

You now have the minimal content needed to run the plugin, so you can activate the plugin in the UI by adding it to `plugins.js`. First, edit the file:

```
$ cd cloudstack/ui/plugins
$ vi plugins.js
```

Now add the following to `plugins.js`:

```
(function($, cloudStack) {
  cloudStack.plugins = [
    'csMyFirstPlugin'
  ];
}(jQuery, cloudStack));
```

6. Check the plugin in the UI.

First, copy all the plugin code that you have created so far to `/tomcat/webapps/client/plugins`. Then refresh the browser and click Plugins in the side navigation bar. You should see your new plugin.

7. Make the plugin do something.

Right now, you just have placeholder content in the new plugin. It's time to add real code. In this example, you will write a basic list view, which renders data from an API call. You will list all virtual machines owned by the logged-in user. To do this, replace the 'show' function in the plugin code with a 'listView' block, containing the required syntax for a list view. To get the data, use the listVirtualMachines API call. Without any parameters, it will return VMs only for your active user. Use the provided 'apiCall' helper method to handle the server call. Of course, you are free to use any other method for making the AJAX call (for example, jQuery's \$.ajax method).

First, open your plugin's JavaScript source file in your favorite editor:

```
$ cd csMyFirstPlugin
$ vi csMyFirstPlugin.js
```

Add the following code in csMyFirstPlugin.js:

```
(function (cloudStack) {
  cloudStack.plugins.csMyFirstPlugin = function(plugin) {
    plugin.ui.addSection({
      id: 'csMyFirstPlugin',
      title: 'My Plugin',
      preFilter: function(args) {
        return isAdmin();
      },

      // Render page as a list view
      listView: {
        id: 'testPluginInstances',
        fields: {
          name: { label: 'label.name' },
          instancename: { label: 'label.internal.name' },
          displayname: { label: 'label.display.name' },
          zonename: { label: 'label.zone.name' }
        },
        dataProvider: function(args) {
          // API calls go here, to retrieve the data asynchronously
          //
          // On successful retrieval, call
          // args.response.success({ data: [data array] });
          plugin.ui.apiCall('listVirtualMachines', {
            success: function(json) {
              var vms = json.listvirtualmachinesresponse.virtualmachine;

              args.response.success({ data: vms });
            },
            error: function(errorMessage) {
              args.response.error(errorMessage)
            }
          });
        }
      }
    });
  };
})(cloudStack);
```

8. Test the plugin.

First, copy all the plugin code that you have created so far to /tomcat/webapps/client/plugins. Then refresh the browser. You can see that your placeholder content was replaced with a list table, containing 4 columns of virtual machine data.

9. Add an action button.

Let's add an action button to the list view, which will reboot the VM. To do this, add an actions block under listView. After specifying the correct format, the actions will appear automatically to the right of each row of data.

```
$ vi csMyFirstPlugin.js
```

Now add the following new code in csMyFirstPlugin.js. (The dots ... show where we have omitted some existing code for the sake of space. Don't actually cut and paste that part):

```
...
listView: {
  id: 'testPluginInstances',
  ...

  actions: {
    // The key/ID you specify here will determine what icon is
    // shown in the UI for this action,
    // and will be added as a CSS class to the action's element
    // (i.e., '.action.restart')
    //
    // -- here, 'restart' is a predefined name in CloudPlatform that will
    // automatically show a 'reboot' arrow as an icon;
    // this can be changed in csMyFirstPlugin.css
    restart: {
      label: 'Restart VM',
      messages: {
        confirm: function() { return 'Are you sure you want to restart this VM?' },
        notification: function() { return 'Rebooted VM' }
      },
      action: function(args) {
        // Get the instance object of the selected row from context
        //
        // -- all currently loaded state is stored in 'context' as objects,
        // such as the selected list view row,
        // the selected section, and active user
        //
        // -- for list view actions, the object's key will be the same as
        // listView.id, specified above;
        // always make sure you specify an 'id' for the listView,
        // or else it will be 'undefined!'
        var instance = args.context.testPluginInstances[0];

        plugin.ui.apiCall('rebootVirtualMachine', {
          // These will be appended to the API request
          //
          // i.e., rebootVirtualMachine&id=...
          data: {
            id: instance.id
          },
          success: function(json) {
            args.response.success({
              // This is an async job, so success here only indicates
              // that the job was initiated.
              //
              // To pass the job ID to the notification UI
              // (for checking to see when action is completed),
              // '_custom: { jobId: ... }' needs to always be passed on success,
              // in the same format as below
              _custom: { jobId: json.rebootvirtualmachineresponse.jobid }
            });
          },
        },
      },
    },
  },
}
```

```
        error: function(errorMessage) {
            args.response.error(errorMessage); // Cancel action, show error message
        returned
        }
    });
},

// Because rebootVirtualMachine is an async job, we need to add
// a poll function, which will periodically check
// the management server to see if the job is ready
// (via pollAsyncJobResult API call)
//
// The plugin API provides a helper function, 'plugin.ui.pollAsyncJob',
// which will work for most jobs
// in CloudPlatform
notification: {
    poll: plugin.ui.pollAsyncJob
}
},

dataProvider: function(args) {
    ...
}
...

```

10. Add the thumbnail icon.

Create an icon file; it should be square, about 50x50 pixels, and named icon.png. Copy it into the same directory with your plugin code: cloudstack/ui/plugins/csMyFirstPlugin/icon.png.

11. Add the stylesheet.

Create a CSS file, with the same name as your .js file. Copy it into the same directory with your plugin code: cloudstack/ui/plugins/csMyFirstPlugin/csMyFirstPlugin.css.

Writing a Storage Plugin

This section gives an outline of how to implement a plugin to integrate a third-party storage provider. For details and an example, you will need to read the code.



Note

Example code is available at: `plugins/storage/volume/sample`

Third party storage providers can integrate with CloudPlatform to provide either primary storage or secondary storage. For example, CloudPlatform provides plugins for Amazon Simple Storage Service (S3) or OpenStack Object Storage (Swift). The S3 plugin can be used for any object storage that supports the Amazon S3 interface.

Additional third party object storages that do not support the S3 interface can be integrated with CloudPlatform by writing plugin software that uses the object storage plugin framework. Several new interfaces are available so that storage providers can develop vendor-specific plugins based on well-defined contracts that can be seamlessly managed by CloudPlatform.

Artifacts such as templates, ISOs and snapshots are kept in storage which CloudPlatform refers to as secondary storage. To improve scalability and performance, as when a number of hosts access secondary storage concurrently, object storage can be used for secondary storage. Object storage can also provide built-in high availability capability. When using object storage, access to secondary storage data can be made available across multiple zones in a region. This is a huge benefit, as it is no longer necessary to copy templates, snapshots etc. across zones as would be needed in an environment using only zone-based NFS storage.

The user enables a storage plugin through the UI. A new dialog box choice is offered to select the storage provider. Depending on which provider is selected, additional input fields may appear so that the user can provide the additional details required by that provider, such as a user name and password for a third-party storage account.

10.1. Overview of How to Write a Storage Plugin

To add a third-party storage option to CloudPlatform, follow these general steps (explained in more detail later in this section):

1. Implement the following interfaces in Java:
 - `DataStoreDriver`
 - `DataStoreLifecycle`
 - `DataStoreProvider`
2. Hardcode your plugin's required additional input fields into the code for the Add Secondary Storage or Add Primary Storage dialog box.
3. Place your `.jar` file in `plugins/storage/volume/` or `plugins/storage/image/`.
4. Edit `/client/tomcatconf/componentContext.xml.in`.
5. Edit `client/pom.xml`.

10.2. Implementing DataStoreDriver

DataStoreDriver contains the code that CloudPlatform will use to provision the object store, when needed.

You must implement the following methods:

- getTO()
- getStoreTO()
- createAsync()
- deleteAsync()

The following methods are optional:

- resize()
- canCopy() is optional. If you set it to true, then you must implement copyAsync().

10.3. Implementing DataStoreLifecycle

DataStoreLifecycle contains the code to manage the storage operations for ongoing use of the storage. Several operations are needed, like create, maintenance mode, delete, etc.

You must implement the following methods:

- initialize()
- maintain()
- cancelMaintain()
- deleteDataStore()
- Implement one of the attach*() methods depending on what scope you want the storage to have: attachHost(), attachCluster(), or attachZone().

10.4. Implementing DataStoreProvider

DataStoreProvider contains the main code of the data store.

You must implement the following methods:

- getDatastoreLifeCycle()
- getDataStoreDriver()
- getTypes(). Returns one or more types of storage for which this data store provider can be used. For secondary object storage, return IMAGE, and for a Secondary Staging Store, return ImageCache.
- configure(). First initialize the lifecycle implementation and the driver implementation, then call registerDriver() to register the new object store provider instance with CloudPlatform.
- getName(). Returns the unique name of your provider; for example, this can be used to get the name to display in the UI.

The following methods are optional:

- `getHostListener()` is optional; it's for monitoring the status of the host.

10.5. Place the .jar File in the Right Directory

For a secondary storage plugin, place your .jar file here:

```
plugins/storage/image/
```

For a primary storage plugin, place your .jar file here:

```
plugins/storage/volume/
```

10.6. Edit Configuration Files

First, edit the following file tell CloudPlatform to include your .jar file. Add a line to this file to tell the CloudPlatform Management Server that it now has a dependency on your code:

```
client/pom.xml
```

Place some facts about your code in the following file so CloudPlatform can run it:

```
/client/tomcatconf/componentContext.xml.in
```

In the section “Deployment configurations of various adapters,” add this:

```
<bean id="some unique ID" class="package name of your implementation of DataStoreProvider">  
  bean</bean>
```

In the section “Storage Providers,” add this:

```
<property name="providers">  
  <ref local="same ID from the bean tag's id attribute">  
</property>
```


Amazon Web Service Interface

11.1. Amazon Web Services EC2 Compatible Interface

CloudPlatform can translate Amazon Web Services (AWS) API calls to native CloudPlatform API calls so that users can continue using existing AWS-compatible tools. This translation service runs as a separate web application in the same tomcat server as the management server of CloudPlatform, listening on the same port. This Amazon EC2-compatible API is accessible through a SOAP web service and the AWS Query API. The AWS Java SDK and AWS PHP SDK are both supported by the Query API.

Limitations:

- Supported only in zones that use basic networking.
- Available in fresh installations of CloudPlatform 3.0.3 and newer. Not available through upgrade of previous versions.
- If you need to support features such as elastic IP, set up a Citrix NetScaler to provide this service. The commands such as `ec2-associate-address` will not work without EIP setup. Users running VMs in this zone will be using the NetScaler-enabled network offering (`DefaultSharedNetscalerEIP` and `ELBNetworkOffering`).

11.2. System Requirements

- This interface complies with Amazon's WDSL version dated August 15, 2012, available at <http://ec2.amazonaws.com/doc/2012-08-15/>.
- Compatible with the EC2 command-line tools *EC2 tools v. 1.6.2.0*, which can be downloaded at <http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.6.2.0.zip>.

11.3. Enabling the AWS API Compatible Interface

The software that provides AWS API compatibility is installed along with CloudPlatform. However, you must enable the feature and perform some setup steps.

1. Set the global configuration parameter `enable.ec2.api` to true. See Setting Configuration Parameters in the CloudPlatform Installation Guide.
2. Create a set of CloudPlatform service offerings with names that match the Amazon service offerings. You can do this through the CloudPlatform UI as described in the Administration Guide.



Warning

Be sure you have included the Amazon default service offering, `m1.small`.

3. If you did not already do so when you set the configuration parameter in step 1, restart the Management Server.

```
# service cloudstack-management restart
```

4. (Optional) The AWS API listens for requests on port 7080. If you prefer AWS API to listen on another port, you can change it as follows:
 - a. Edit the files `/etc/cloudstack/management/server.xml`, `/etc/cloudstack/management/server-nonssl.xml`, and `/etc/cloudstack/management/server-ssl.xml`.
 - b. In each file, find the tag `<Service name="Catalina7080">`. Under this tag, locate `<Connector executor="tomcatThreadPool-internal" port= ... >`.
 - c. Change the port to whatever port you want to use, then save the files.
 - d. Restart the Management Server.



Note

If you re-install CloudPlatform, you will have to make these changes again.

11.4. AWS API User Setup Steps (SOAP Only)

In general, users need not be aware that they are using a translation service provided by CloudPlatform. They need only send AWS API calls to CloudPlatform's endpoint, and it will translate the calls to the native API. Users of the Amazon EC2 compatible interface will be able to keep their existing EC2 tools and scripts and use them with their CloudPlatform deployment, by specifying the endpoint of the management server and using the proper user credentials. In order to do this, each user must perform the following configuration steps:

- Generate user credentials and register with the service.
- Set up the environment variables for the EC2 command-line tools.
- For SOAP access, use the endpoint `http://CloudPlatform-management-server:7080/awsapi`. The `CloudPlatform-management-server` can be specified by a fully-qualified domain name or IP address.

11.4.1. AWS API User Registration

Each user must perform a one-time registration. The user follows these steps:

1. Obtain the following by looking in the CloudPlatform UI, using the API, or asking the cloud administrator:
 - The CloudPlatform server's publicly available DNS name or IP address
 - The user account's API key and Secret key
2. Generate a private key and a self-signed X.509 certificate. The user substitutes their own desired storage location for `/path/to/...` below.

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /path/to/private_key.pem -  
out /path/to/cert.pem
```

3. Register the mapping from the X.509 certificate to the API/Secret keys. Download the following script from <http://download.cloud.com/releases/3.0.6/cloudstack-aws-api-register> and run it. Substitute the values that were obtained in step 1 in the URL below.

```
$ cloudstack-aws-api-register --apikey=User's CloudPlatform API key --
secretkey=User's CloudPlatform Secret key --cert=/path/to/cert.pem --
url=http://CloudPlatform.server:7080/awsapi
```



Note

A user with an existing AWS certificate could choose to use the same certificate with CloudPlatform, but the public key would be uploaded to the CloudPlatform management server database.

11.4.2. AWS API Command-Line Tools Setup

To use the EC2 command-line tools, the user must perform these steps:

1. Be sure you have the right version of EC2 Tools. The supported version is available at <http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.6.2.0.zip>.
2. Set up the environment variables that will direct the tools to the server. As a best practice, you may wish to place these commands in a script that may be sourced before using the AWS API translation feature.

```
$ export EC2_CERT=/path/to/cert.pem
$ export EC2_PRIVATE_KEY=/path/to/private_key.pem
$ export EC2_URL=http://CloudPlatform.server:7080/awsapi
$ export EC2_HOME=/path/to/EC2_tools_directory
```

11.5. Supported AWS API Calls

The following Amazon EC2 commands are supported by CloudPlatform when the AWS API compatibility feature is enabled. For a few commands, there are differences between the CloudPlatform and Amazon EC2 versions, and these differences are noted. The underlying SOAP / REST call for each command is also given, for those who have built tools using those calls.

Table 11.1. Elastic IP

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-allocate-address	AllocateAddress	associateIpAddress
ec2-associate-address	AssociateAddress	enableStaticNat
ec2-describe-addresses	DescribeAddresses	listPublicIpAddresses
ec2-disassociate-address	DisassociateAddress	disableStaticNat
ec2-release-address	ReleaseAddress	disassociateIpAddress

Table 11.2. Availability Zone

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-describe-availability-zones	DescribeAvailabilityZones	listZones

Table 11.3. Images

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-create-image	CreateImage	createTemplate

EC2 command	SOAP / REST call	CloudPlatform API call
The noReboot parameter is not supported.		
ec2-deregister	DeregisterImage	DeleteTemplate
ec2-describe-images	DescribeImages	listTemplates
ec2-register For the optional parameter architecture , use the CloudPlatform format rather than the EC2 format. The CloudPlatform format includes the template format, zone, OS type, hypervisorm and required parameters. For example, architecture='VHD:basiczone1:Centos 5.3 (64-bit):xenserver'.	RegisterImage	registerTemplate

Table 11.4. Image Attributes

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-describe-image-attribute	DescribeImageAttribute	listTemplatePermissions
ec2-modify-image-attribute	ModifyImageAttribute	updateTemplatePermissions
ec2-reset-image-attribute	ResetImageAttribute	updateTemplatePermissions

Table 11.5. Instances

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-describe-instances In addition to the EC2 VM states, can also return a CloudPlatform error state. Known issue: The CloudPlatform device ID of 0, which represents a root volume, does not map to any EC2 device name to be returned in the command response.	DescribeInstances	listVirtualMachines
ec2-reboot-instances	RebootInstances	rebootVirtualMachine
ec2-run-instances	RunInstances	deployVirtualMachine
ec2-start-instances	StartInstances	startVirtualMachine
ec2-stop-instances	StopInstances	stopVirtualMachine
ec2-terminate-instances	TerminateInstances	destroyVirtualMachine

Table 11.6. Instance Attributes

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-describe-instance-attribute In addition to the EC2 VM states, can also return a CloudPlatform error state.	DescribeInstanceAttribute	listVirtualMachines

EC2 command	SOAP / REST call	CloudPlatform API call
Known issue: The CloudPlatform device ID of 0, which represents a root volume, does not map to any EC2 device name to be returned in the command response.		

Table 11.7. Keys Pairs

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-add-keypair	CreateKeyPair	createSSHKeyPair
ec2-delete-keypair	DeleteKeyPair	deleteSSHKeyPair
ec2-describe-keypairs	DescribeKeyPairs	listSSHKeyPairs
ec2-import-keypair	ImportKeyPair	registerSSHKeyPair

Table 11.8. Passwords

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-get-password	GetPasswordData	getVMPassWord

Table 11.9. Security Groups

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-authorize	AuthorizeSecurityGroupIngress	authorizeSecurityGroupIngress
ec2-add-group	CreateSecurityGroup	createSecurityGroup
ec2-delete-group	DeleteSecurityGroup	deleteSecurityGroup
ec2-describe-group	DescribeSecurityGroups	listSecurityGroups
ec2-revoke	RevokeSecurityGroupIngress	revokeSecurityGroupIngress

Table 11.10. Snapshots

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-create-snapshot	CreateSnapshot	createSnapshot
ec2-delete-snapshot	DeleteSnapshot	deleteSnapshot
ec2-describe-snapshots	DescribeSnapshots	listSnapshots

Table 11.11. Volumes

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-attach-volume	AttachVolume	attachVolume
ec2-create-volume	CreateVolume	createVolume
ec2-delete-volume	DeleteVolume	deleteVolume
ec2-describe-volume	DescribeVolumes	listVolumes
ec2-detach-volume	DetachVolume	detachVolume

Table 11.12. Resource Tags

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-create-tags	CreateTags	Add tags to one or more resources.

EC2 command	SOAP / REST call	CloudPlatform API call
ec2-delete-tags	DeleteTags	Remove tags from one or more resources.
ec2-describe-tags	DescribeTags	Show currently defined tags.

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 the listAlerts API command.

```
MEMORY = 0 // Available Memory below configured threshold
```

```
CPU = 1 // Unallocated CPU below configured threshold
```

```
STORAGE =2 // Available Storage below configured threshold
```

```
STORAGE_ALLOCATED = 3 // Remaining unallocated Storage is below configured threshold
```

```
PUBLIC_IP = 4 // Number of unallocated virtual network public IPs is below configured threshold
```

```
PRIVATE_IP = 5 // Number of unallocated private IPs is below configured threshold
```

```
HOST = 6 // Host related alerts like host disconnected
```

```
USERVM = 7 // User VM stopped unexpectedly
```

```
DOMAIN_ROUTER = 8 // Domain Router VM stopped unexpectedly
```

```
CONSOLE_PROXY = 9 // Console Proxy VM stopped unexpectedly
```

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

```
STORAGE_MISC = 11 // Storage issue in system VMs
```

```
USAGE_SERVER = 12 // No usage server process running
```

```
MANAGMENT_NODE = 13 // Management network CIDR is not configured originally
```

```
DOMAIN_ROUTER_MIGRATE = 14 // Domain Router VM Migration was unsuccessful
```

```
CONSOLE_PROXY_MIGRATE = 15 // Console Proxy VM Migration was unsuccessful
```

```
USERVM_MIGRATE = 16 // User VM Migration was unsuccessful
```

```
VLAN = 17 // Number of unallocated VLANs is below configured threshold in availability zone
```

```
SSVM = 18 // SSVM stopped unexpectedly
```

```
USAGE_SERVER_RESULT = 19 // Usage job failed
```

Appendix B. Alerts

STORAGE_DELETE = 20 // Failed to delete storage pool

UPDATE_RESOURCE_COUNT = 21 // Failed to update the resource count

USAGE_SANITY_RESULT = 22 // Usage Sanity Check failed

DIRECT_ATTACHED_PUBLIC_IP = 23 // Number of unallocated shared network IPs is low in availability zone

LOCAL_STORAGE = 24 // Remaining unallocated Local Storage is below configured threshold

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