



CloudBridge Guide

Version 1.0.1

Revised August 24, 2011



© 2011 Citrix Systems, Inc. All rights reserved. Specifications are subject to change without notice. The Cloud.com logo, Cloud.com, and CloudStack are trademarks or registered trademarks of Citrix Systems, Inc. All other brands or products are trademarks or registered trademarks of their respective holders.

Contents

1	Overview	5
2	System Requirements	6
3	Deployment Architecture	7
4	Installing CloudBridge	8
4.1	Operating System and OS Preparation	8
4.2	Installing the First CloudBridge Server	8
4.3	Installing Additional CloudBridge Servers (Optional)	10
4.4	Setting Up Database Replication (Optional)	11
4.4.1	Failover	13
5	User Setup	14
5.1	User Registration	14
5.2	Endpoints	14
5.3	Using the ec2-api-tools	14
6	Best Practices	16
6.1	Ensuring Command Completion: Timeouts	16
7	Supported EC2 Commands	17
8	CloudBridge Utility Commands	21
8.1	CloudEC2Version	21
8.2	SetUserKeys	21
8.3	SetCertificate	21
8.4	DeleteCertificate	22
8.5	SetOfferMapping	22
8.6	DeleteOfferMapping	22
9	Unsupported Commands	23
9.1	EC2 AMI Tools	23
9.2	EC2 API Commands	23



10	Troubleshooting.....	24
10.1	A command failed. Where are the logs?	24
10.2	CloudBridge can't connect to CloudStack	24
11	More Information	25

1 Overview

CloudBridge is a server process that runs as a companion to CloudStack. CloudBridge provides an Amazon EC2 compatible API accessible through both SOAP and REST web services. The EC2 API calls are translated to CloudStack API calls by CloudBridge. Clients can continue using existing EC2-compatible tools.

CloudBridge requires a RHEL5/CentOS server that is separate from the CloudStack Management Server. CloudBridge connects to the CloudStack Management Server to implement the user requests.



2 System Requirements

CloudBridge requires the following hardware and software.

<p>CloudBridge Server</p>	<p>Hosts CloudBridge software</p>	<p>Minimum requirements:</p> <ul style="list-style-type: none"> • 64-bit x86 CPU (more cores results in better performance) • 2 GB of memory • 80 GB of local disk • At least 1 NIC • RHEL/CentOS 5.4+ 64-bit • Statically allocated IP address • Fully qualified domain name as returned by the hostname command • Should not be the same host that is running the CloudStack Management Server
<p>EC2 tools v. 1.3.6230</p>	<p>Client interface to the Amazon EC2 service</p>	<p>CloudBridge works with version 1.3.6230 of the EC2 Tools. Download the correct version here: http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-62308.zip</p>

Complies with Amazon's WDSL version dated November 15, 2010, available at <http://ec2.amazonaws.com/doc/2010-11-15/>.

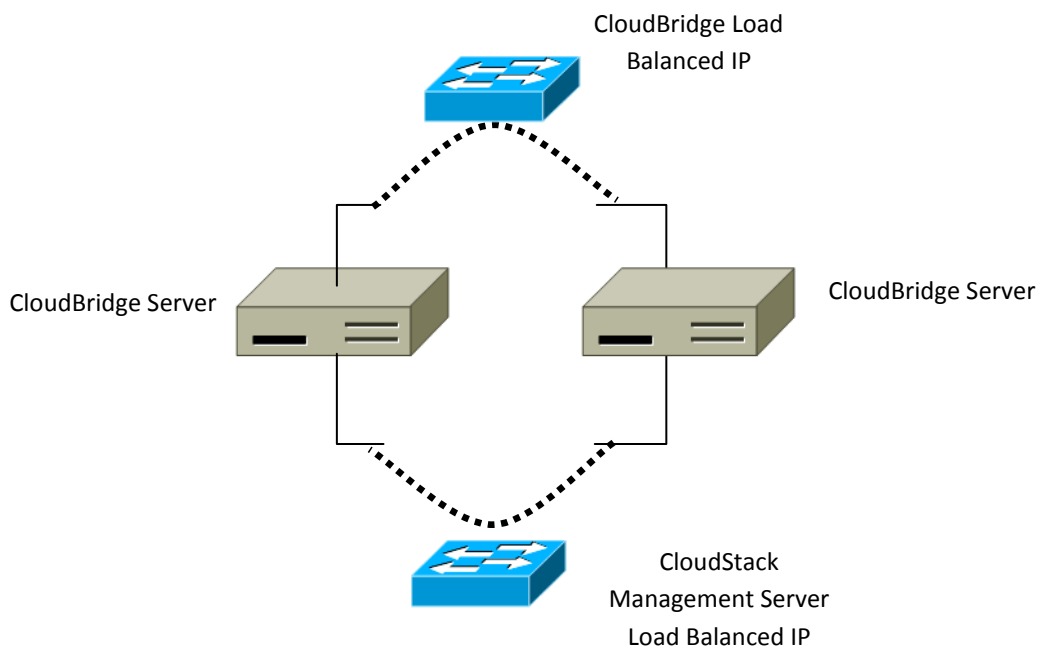
3 Deployment Architecture

CloudBridge architecture varies depending on the size and purpose of the deployment.

All CloudBridge deployments include a MySQL server to store credential mappings. The CloudBridge API client sets up a mapping between a user's EC2 certificate and CloudStack API keys. This mapping is stored in the MySQL database.

All CloudBridge deployments include at least one CloudBridge server. The CloudBridge server runs in the Tomcat container. Trial deployments may use a single CloudBridge server. Production deployments generally use two CloudBridge servers behind a load balancer. The load balancer need not establish sticky sessions for CloudBridge access. The CloudBridge server listens on port 8090 by default. You may map port 80 of a public IP address to port 8090 of your CloudBridge servers.

CloudBridge can access the CloudStack Management Server through a public or private IP address. If you have multiple CloudStack Management Servers, configure a load-balanced IP address that CloudBridge can use to access the CloudStack Management Servers. By default, CloudBridge will use port 8080 to communicate with the CloudStack Management Server.



4 Installing CloudBridge

The CloudBridge Server download includes everything you need to get started, except MySQL. This includes the Cloud.com software as well as dependencies. This section describes installing one or more CloudBridge Servers with one instance of MySQL, which may be on a different node from the CloudBridge Servers.

Summary of installation steps:

1. Prepare the operating system for all CloudBridge servers.
2. Install the CloudBridge Server.
3. Install MySQL.
4. (Optional) Install additional CloudBridge Servers to create a farm for high availability.
5. (Optional) Set up database replication.

4.1 Operating System and OS Preparation

The CloudBridge Server requires RHEL/CentOS 5.4 64 bit or later. You can download CentOS 64-bit from http://isoredirect.centos.org/centos/5/isos/x86_64/. The OS must be prepared to host the CloudBridge Server using the following steps.

Important: These steps must be done on all CloudBridge Servers.

1. Edit the `/etc/hosts` file to make sure that every CloudBridge Server has a fully-qualified host name that resolves to an IP address. Alternatively, you can do this through DNS.
2. Log in as root.
3. Ensure that the SELINUX variable in `/etc/selinux/config` is set to permissive. This ensures that MySQL and the CloudBridge Server can run properly on system reboot.
4. Run the following command.

```
# setenforce permissive
```

5. Make sure that the CloudBridge Server can reach the Internet.

```
# ping www.google.com
```

6. Open up port 8090 on the CloudBridge server.

```
# iptables -I INPUT -p tcp --dport 8090 -j ACCEPT
```

7. Edit the `/etc/sysconfig/iptables` file and add the following lines at the beginning of the INPUT chain. This will keep 8090 open on host reboot.

```
-A INPUT -p tcp --dport 8090 -j ACCEPT
```

4.2 Installing the First CloudBridge Server

This section tells how to install a CloudBridge server. These steps install the following:

- CloudBridge Server software is installed under `/usr/share/cloud/bridge`.

- Configuration settings for CloudBridge are placed in the file `/usr/share/cloud/bridge/conf/ec2-service.properties`, and other Tomcat configuration can be found in the same directory.

If you plan to scale your installation to include multiple CloudBridge Servers in a load-balanced pool, follow the instructions in this section for the first server, then go to [Installing Additional CloudBridge Servers \(Optional\)](#) on page 10.

1. Be sure the steps in [Operating System and OS Preparation](#) on page 8 have been performed.
2. Install Java and Tomcat.

```
# yum install java
# cd /etc/yum.repos.d
# wget 'http://www.jpacage.org/jpacage50.repo'
# yum install tomcat6 tomcat6-webapps tomcat6-admin-webapps
```

3. Install MySQL. First choose where to install it, then run the commands below on that machine.
 - In a single-node system, you can install MySQL on the same machine where you just installed Java and Tomcat.
 - For a multi-node system, you will more likely install the MySQL server on a separate node. It is also permitted to have MySQL installed on the same node with a CloudBridge Server.

```
# yum install mysql mysql-server
# service mysqld start
```

4. Install the CloudBridge package. You should have a file in the form of “cloud-bridge-xxx.rpm:

```
# rpm -ivh cloud-bridge-xxx.rpm
```

5. Set up the bridge's configuration. In this step you will create a mapping from common EC2 service offerings to CloudStack service offerings. CloudStack service offerings are referred to by their Service Offering IDs. These IDs are available in the CloudStack Admin UI.

```
# cloud-setup-bridge
Welcome to the CloudBridge setup.
Enter suitable values or press enter for default.

Management server hostname or IP [127.0.0.1]: 92.52.146.124
Management server port [8080]: 80
Service offering ID for m1.small [1]:
Service offering ID for m1.large [2]:
Service offering ID for m1.xlarge [4]: 2
Service offering ID for c1.medium [3]: 2
Service offering ID for c1.xlarge [3]: 2
Service offering ID for m2.xlarge [3]: 2
Service offering ID for m2.2xlarge [3]: 2
Service offering ID for m2.4xlarge [3]: 2
Service offering ID for cc1.4xlarge [3]: 2

Values saved. Restart the cloud-bridge service for the changes to become active.
```

6. Set up the CloudBridge database schema and start the CloudBridge service.

```
# cloud-setup-bridge-db
# service cloud-bridge start
```

7. Each CloudBridge user must perform a one-time registration process. Registration should be done once for each user, whether you intend to install one CloudBridge server or multiple CloudBridge servers. This is a good time to register yourself as a user, following the steps in User Setup on page 14.

After this step, the installation of CloudBridge Server on one node is complete, and a user is registered with CloudBridge. You can start issuing test commands to the server, or continue with the next step to add some optional features to your installation.

8. You can add to the installation with the following procedures:
 - Installing Additional CloudBridge Servers (Optional) on page 10
 - Setting Up Database Replication (Optional) on page 11

4.3 Installing Additional CloudBridge Servers (Optional)

This section tells how to add another CloudBridge server to an existing set of one or more CloudBridge servers in a load balanced pool.

1. Be sure the following preparation steps have been performed:
 - The first CloudBridge node and MySQL have been installed. See Installing the First CloudBridge Server on page 8.
 - For each additional node, perform Operating System and OS Preparation on page 8.

2. Install Java and Tomcat on the new node.

```
# yum install java
# cd /etc/yum.repos.d
# wget 'http://www.jpackage.org/jpackage50.repo'
# yum install tomcat6 tomcat6-webapps tomcat6-admin-webapps
```

3. Install the CloudBridge package. You should have a file in the form of "cloud-bridge-xxx.rpm".

```
# rpm -ivvh cloud-bridge-xxx.rpm
```

4. Configure CloudBridge. For these steps, provide the same answers as for the first CloudBridge server.

```
# cloud-setup-bridge
Welcome to the CloudBridge setup.
Enter suitable values or press enter for default.

Management server hostname or IP [127.0.0.1]: 92.52.146.124
Management server port [8080]: 80
Service offering ID for m1.small [1]:
Service offering ID for m1.large [2]:
Service offering ID for m1.xlarge [4]: 2
Service offering ID for c1.medium [3]: 2
Service offering ID for c1.xlarge [3]: 2
Service offering ID for m2.xlarge [3]: 2
Service offering ID for m2.2xlarge [3]: 2
Service offering ID for m2.4xlarge [3]: 2
Service offering ID for cc1.4xlarge [3]: 2

Values saved. Restart the cloud-bridge service for the changes to become active.
```

5. Edit the configuration for the new server to use the MySQL instance that was installed with the first CloudBridge Server. Edit the file `/usr/share/cloud/bridge/conf/ec2-service.properties` and set `dbHost` to the hostname of the MySQL server. Then edit `/usr/share/cloud/bridge/conf/hibernate.cfg.xml` and set `hibernate.connection.url` to the hostname of the MySQL server.

```
# cd /usr/share/cloud/bridge/conf
# vi ec2-service.properties
# vi hibernate.cfg.xml
```

8. Finally, restart the cloud-bridge service.

```
# service cloud-bridge restart
```

Your additional CloudBridge server is now available for service.

4.4 Setting Up Database Replication (Optional)

CloudBridge supports database replication from one MySQL node to another. This is achieved using standard MySQL replication. You may want to do this as insurance against MySQL server or storage loss. If the main database instance fails, the CloudBridge administrator can manually fail over to the replicated database (see Failover on page 13).

MySQL replication is implemented using a master/slave model. The master is the node that the CloudBridge Servers are configured to use. The slave is a standby node that receives all write operations from the master and applies them to a local, redundant copy of the database.

Important: These steps assume that this is a fresh install with no data in the master.

Important: Creating a replica is not a backup solution. You should develop a backup procedure for the MySQL data that is distinct from replication.

1. Edit the MySQL configuration (`/etc/my.cnf` or `/etc/mysql/my.cnf`, depending on your OS) on the master and add the following in the `[mysqld]` section below `datadir`.

```
log_bin=mysql-bin
server_id=1
```

The `server_id` must be unique with respect to other servers. A common practice is to set `server_id` to the last octet of the server's IP address.

2. Restart the MySQL service on the master:

```
# service mysqld restart
```

3. Create a replication account on the master and give it privileges. The following example uses the "cloud-repl" user with the password "password". This assumes that master and slave run on the 172.16.1.0/24 network.

```
# mysql -u root
mysql> create user 'cloud-repl'@'172.16.1.%' identified by 'password';
mysql> grant replication slave on *.* TO 'cloud-repl'@'172.16.1.>';
mysql> flush privileges;
mysql> flush tables with read lock;
```

4. Leave the current MySQL session running.
5. In a new shell, start a second MySQL session.

- Retrieve the current position of the database.

```
# mysql -u root
mysql> show master status;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      412 |              |                  |
+-----+-----+-----+-----+
```

- Note the file and the position that are returned by your instance.
- Exit from this session.
- Complete the master setup. Returning to your first session on the master, release the locks and exit MySQL.

```
mysql> unlock tables;
```

- If you have not already installed MySQL on a second server, do so now. On the second server, run the following commands.

```
# yum install mysql-server
# chkconfig mysqld on
```

- On the slave server, edit my.cnf and add the following lines in the [mysqld] section below datadir.

```
server_id=2
innodb_rollback_on_timeout=1
innodb_lock_wait_timeout=600
```

- Restart MySQL on the slave.

```
# service mysqld restart
```

- Instruct the slave to connect to and replicate from the master. Replace the IP address, password, log file, and position with the values you have used in the previous steps.

```
mysql> change master to
-> master_host='172.16.1.217',
-> master_user='cloud-repl',
-> master_password='password',
-> master_log_file='mysql-bin.000001',
-> master_log_pos=412;
```

- Start replication on the slave.

```
mysql> start slave;
```

- Optionally, open port 3306 on the slave as was done on the master earlier.

Important: This is not required for replication to work. But if you choose not to do this, you will need to do it when failover to the slave occurs.

- Optionally, repeat steps 10 through 15 to add more slaves.

4.4.1 Failover

Failover from one MySQL instance to another is performed by the administrator. In the event of a database failure, perform the following steps.

1. Stop the CloudBridge Servers by running the following command on each server node.

```
# service cloud-bridge stop
```

2. Reconfigure a slave server to be the new master. If you have additional slaves available, configure them to receive replicated data from the new master. Follow the steps in [Setting Up Database Replication \(Optional\)](#) on page 11.
3. Ensure that the new master's port 3306 is open to the CloudBridge Servers.
4. Edit the configuration of each CloudBridge Server to use the new master. Edit the file `/usr/share/cloud/bridge/conf/ec2-service.properties` and set `dbHost` to the hostname of the MySQL server. Then edit `/usr/share/cloud/bridge/conf/hibernate.cfg.xml` and set `hibernate.connection.url` to the hostname of the MySQL server.

```
# cd /usr/share/cloud/bridge/conf
# vi ec2-service.properties
# vi hibernate.cfg.xml
```

5. Restart the CloudBridge Servers by running the following command on each server node.

```
# service cloud-bridge restart
```

5 User Setup

In general, CloudBridge users need not be aware that they are using a CloudBridge server. They need only execute EC2 API calls to CloudBridge and it will translate the calls to CloudStack's native API. However, each user must perform the following setup steps:

- Register with CloudBridge. See 5.1 User Registration.
- Set up their environment and/or tools appropriately to use the CloudBridge endpoint. See 5.2 Endpoints and 5.3 Using the ec2-api-tools.

5.1 User Registration

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

1. Obtain the following from your CloudStack cloud administrator:
 - The CloudBridge server's publicly available DNS name or IP address
 - Your account's API key and Secret key
2. Generate a private key and a self-signed X.509 certificate. Substitute your 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 in CloudBridge. Substitute the values you obtained from the CloudStack administrator in the URL below.

```
$ cloud-bridge-register --apikey=<User's Cloudstack API key>  
--secretkey=<User's CloudStack Secret key> --cert=</path/to/cert.pem>  
--url=http://<cloud-bridge-server>:8090/bridge
```

5.2 Endpoints

For SOAP access, the endpoint is:

```
http://<fqdn-or-ip>:<port>/bridge/services/AmazonEC2
```

For REST access, the endpoint is:

```
http://<fqdn-or-ip>:<port>/bridge/rest/AmazonEC2
```

5.3 Using the ec2-api-tools

To enable the Amazon EC2 API tools to work through CloudBridge, the user must perform these steps:

1. Register the certificate and keys with CloudBridge (see User Registration on page 14).
2. Be sure you have the version of EC2 Tools that works with CloudBridge. The supported version is available at <http://s3.amazonaws.com/ec2-downloads/ec2-api-tools-1.3-62308.zip>.

3. Set up the environment variables that will direct the tools to the CloudBridge server. As a best practice, you may wish to place these commands in a script that may be sourced before using CloudBridge.

```
$ export EC2_ACCESS_KEY=<CloudStack API key>
$ export EC2_SECRET_KEY=<CloudStack Secret key>
$ export EC2_CERT=/path/to/cert.pem
$ export EC2_PRIVATE_KEY=/path/to/private_key.pem
$ export EC2_URL=http://cloud-bridge-hostname:8090/bridge
```

6 Best Practices

6.1 Ensuring Command Completion: Timeouts

The Amazon EC2 command-line tools have a default connection timeout. When used with CloudBridge, a longer timeout might be needed for some commands. If you find that commands are not completing due to timeouts, you can gain more time for commands to finish by overriding the default timeouts on individual commands. You can add the following optional command-line parameters to any CloudBridge-supported EC2 command:

<code>--connection-timeout <i>TIMEOUT</i></code>	Specifies a connection timeout (in seconds). Example: <code>--connection-timeout 30</code>
<code>--request-timeout <i>TIMEOUT</i></code>	Specifies a request timeout (in seconds). Example: <code>--request-timeout 45</code>

Example:

```
ec2-run-instances 2 -z us-test1 -n 1-3 -t --connection-timeout 120 --request-timeout 120
```

7 Supported EC2 Commands

The following Amazon EC2 commands are supported by CloudBridge. For a few commands, there are differences between the CloudBridge and Amazon EC2 versions, and these differences are noted. The underlying SOAP call for each command is also given, for those who have built tools using those calls.

Elastic IP Addresses

EC2 Command	SOAP Call
ec2-allocate-address	AllocateAddress
ec2-associate-address	AssociateAddress
ec2-describe-addresses	DescribeAddresses
ec2-disassociate-address	DisassociateAddress
ec2-release-address	ReleaseAddress

Availability Zones

EC2 Command	SOAP Call
ec2-describe-availability-zones	DescribeAvailabilityZones

Images

EC2 Command	SOAP Call
ec2-create-image	CreateImage
ec2-deregister	DeregisterImage
ec2-describe-images	DescribeImages
ec2-register <ul style="list-style-type: none"> In CloudBridge, the <code>architecture</code> parameter is required and is used to pass three required values: the template format (QCOW2, RAW, or VHD); zone where the template is hosted; and template OS. Use the format "<code><format>:<zoneName>:<osTypeName></code>". For example, "VHD:ZONE1:Centos 4.5" The <code>imageLocation</code> parameter is the URL where the template is hosted, starting with <code>http://</code> or <code>https://</code>. 	RegisterImage

Image Attributes

EC2 Command	SOAP Call
ec2-describe-image-attribute (partially supported)	DescribeImageAttribute
ec2-modify-image-attribute (partially supported)	ModifyImageAttribute
ec2-reset-image-attribute	ResetImageAttribute



Instances

EC2 Command	SOAP Call
ec2-describe-instances <ul style="list-style-type: none"> The following filters are supported: <ul style="list-style-type: none"> availability-zone hypervisor image-id instance-id instance-type instance-state-code instance-state-name ip-address owner-id root-device-name 	DescribeInstances
ec2-run-instances <ul style="list-style-type: none"> Requires the <code>--availability-zone (-z)</code> parameter. Requires both min and max number of instances in the <code>--instance-count (-n)</code> parameter. To refer to the desired template, provide the template ID in the <code>--instance-type (-t)</code> parameter. 	RunInstances
ec2-reboot-instances	RebootInstances
ec2-start-instances	StartInstances
ec2-stop-instances	StopInstances
ec2-terminate-instances	TerminateInstances

Instance Attributes

EC2 Command	SOAP Call
ec2-describe-instance-attribute <ul style="list-style-type: none"> Partially supported. Only the <code><instanceId> -t</code> options are supported. 	DescribeInstanceAttribute
ec2-modify-instance-attribute (partially supported)	ModifyInstanceAttribute

Key Pairs

EC2 Command	SOAP Call
ec2-add-keypair	CreateKeyPair
ec2-delete-keypair	DeleteKeyPair
ec2-describe-keypairs	DescribeKeyPairs
ec2-import-keypair	ImportKeyPair

Passwords

EC2 Command	SOAP Call
ec2-get-password	GetPasswordData

Security Groups

EC2 Command	SOAP Call
ec2-authorize	AuthorizeSecurityGroupIngress
ec2-add-group	CreateSecurityGroup
ec2-delete-group	DeleteSecurityGroup
ec2-describe-group <ul style="list-style-type: none"> The following filters are supported: <ul style="list-style-type: none"> description group-id group-name ip-permission.cidr ip-permission.from-port ip-permission.to-port ip-permission.protocol owner-id 	DescribeSecurityGroups
ec2-revoke	RevokeSecurityGroupIngress

Snapshots

EC2 Command	SOAP Call
ec2-create-snapshot	CreateSnapshot
ec2-delete-snapshot	DeleteSnapshot
ec2-describe-snapshots <ul style="list-style-type: none"> The following filters are supported: <ul style="list-style-type: none"> owner-alias owner-id (use the CloudStack API key) snapshot-id start-time status volume-id volume-size 	DescribeSnapshots
ec2-modify-snapshot-attribute (partially supported)	ModifySnapshotAttribute

Volumes

EC2 Command	SOAP Call
ec2-attach-volume	AttachVolume
ec2-create-volume <ul style="list-style-type: none"> Must have at least one disk offering with customizable disk size available. 	CreateVolume
ec2-delete-volume	DeleteVolume
ec2-describe-volumes <ul style="list-style-type: none"> The following filters are supported: <ul style="list-style-type: none"> attachment.attach-time attachment.device attachment.instance-id availability-zone create-time size snapshot-id status volume-id 	DescribeVolumes
ec2-detach-volume	DetachVolume



8 CloudBridge Utility Commands

In addition to the Amazon EC2 commands, CloudBridge provides a set of commands for the following configuration tasks.

- Show the EC2 version. See `CloudEC2Version` on page 21.
- Register your Cloud.com account with the Amazon EC2 service. See `SetUserKeys` on page 21.
- Register X.509 certificate in order to use the EC2 SOAP API. See `SetCertificate` on page 21 and `DeleteCertificate` on page 22.
- Map Amazon instance type strings to CloudStack service offering IDs. See `SetOfferMapping` on page 22 and `DeleteOfferMapping` on page 22.

The user keys, certificate, and offer mapping are normally configured during installation, using the setup scripts. You only need to use the utility commands in case you want to modify the configuration later.

8.1 CloudEC2Version

Gives the release version of the EC2 software.

Syntax:

```
http://<fqdn-or-ip>:<port>/bridge/rest/AmazonEC2?Action=CloudEC2Version
```

Returns:

- Example XML response:

```
<CloudEC2Version>1.01</CloudEC2Version>
```

8.2 SetUserKeys

Gives a Cloud.com user's API access and secret keys to the EC2 service so that the EC2 service can call the CloudStack API on behalf of the Cloud.com user. API and secret keys are required for both REST and SOAP access.

- The parameters `accesskey` and `secretkey` are required.
- `SetUserKeys` can be called repeatedly. Subsequent calls overwrite any previously stored values.

Syntax:

```
https://<fqdn-or-ip>:<port>/bridge/rest/  
AmazonEC2?Action=SetUserKeys&accesskey=<key>&secretkey=<key>
```

8.3 SetCertificate

Registers the Cloud.com user's X.509 certificate with the EC2 service. This is required only for SOAP access. EC2 requires the client to have a public/private key pair with the public key defined by a X.509 certificate.

- Always call `SetUserKeys` before `SetCertificate`.
- The parameters `AWSAccessKeyID` and `cert` are required.
- `SetCertificate` can be called repeatedly. Subsequent calls overwrite any previously stored values.

Syntax:

```
http://<fqdn-or-ip>:<port>/bridge/rest/  
AmazonEC2?Action=SetCertificate&AWSAccessKeyId=<Cloud.com API AccessKey>  
&cert=<pem encoded X.509 cert>
```

8.4 DeleteCertificate

Removes the Cloud.com user's X.509 certificate that was previously registered with the EC2 service.

- The parameter AWSAccessKeyId is required and must contain the same ID given in a previous call to SetCertificate.

Syntax:

```
http://<fqdn-or-ip>:<port>/bridge/rest/  
AmazonEC2?Action=DeleteCertificate&AWSAccessKeyId=<Cloud.com API AccessKey>
```

8.5 SetOfferMapping

Maps Amazon instance type strings (such as m1.small, cc1.4xlarge) to CloudStack service offering IDs. Issue one command for each pair. The mapping is initially created through onscreen prompts during CloudBridge setup. The SetOfferMapping command is useful if you want to override the original map later.

- The parameters amazonoffer and cloudoffer are required.

Syntax:

```
http://<fqdn-or-ip>:<port>/bridge/rest/AmazonEC2?Action=SetOfferMapping  
&amazonoffer=<Amazon instance type>&cloudoffer=<Service offering ID>
```

8.6 DeleteOfferMapping

Removes a mapping that was previously created either through the CloudBridge setup script or a call to SetOfferMapping.

- The parameter amazonoffer is required.

Syntax:

```
http://<fqdn-or-ip>:<port>/bridge/rest/AmazonEC2?Action=SetOfferMapping  
&amazonoffer=<Amazon instance type>
```

9 Unsupported Commands

The following Amazon EC2 commands are not currently supported by CloudBridge. As an open source project, CloudBridge invites you to help implement these commands. See <http://cloudstack.org/about-cloudstack/contribute.html>.

9.1 EC2 AMI Tools

None of the Amazon EC2 AMI Tools are supported. See <http://aws.amazon.com/developertools/368>.

9.2 EC2 API Commands

The following Amazon EC2 API commands are not supported.

ec2-activate-license	ec2-delete-customer-gateway	ec2-describe-spot-instance-requests
ec2-associate-dhcp-options	ec2-delete-dhcp-options	ec2-describe-spot-price-history
ec2-attach-vpn-gateway	ec2-delete-placement-group	ec2-describe-subnets
ec2-bundle-instance	ec2-delete-spot-datafeed-subscription	ec2-describe-subnets
ec2-cancel-bundle-task	ec2-delete-subnet	ec2-describe-tags
ec2-cancel-conversion-task	ec2-delete-vpn-connection	ec2-describe-vpcs
ec2-cancel-spot-instance-requests	ec2-delete-vpn-gateway	ec2-describe-vpn-connections
ec2-confirm-product-instance	ec2-describe-bundle-tasks	ec2-describe-vpn-gateways
ec2-create-customer-gateway	ec2-describe-conversion-tasks	ec2-detach-vpn-gateway
ec2-create-dhcp-options	ec2-describe-customer-gateways	ec2-get-console-output
ec2-create-placement-group	ec2-describe-dhcp-options	ec2-import-instance
ec2-create-spot-datafeed-subscription	ec2-describe-licenses	ec2-import-volume
ec2-create-subnet	ec2-describe-placement-groups	ec2-purchase-reserved-instances-offering
ec2-create-tags	ec2-describe-reserved-instances	ec2-request-spot-instances
ec2-create-vpc	ec2-describe-reserved-instances-offerings	ec2-reset-instance-attribute
ec2-create-vpn-connection	ec2-describe-snapshot-attribute	ec2-reset-snapshot-attribute
ec2-create-vpn-gateway	ec2-describe-spot-datafeed-subscription	ec2-reset-image-attribute
ec2-deactivate-license		

10 Troubleshooting

10.1 A command failed. Where are the logs?

CloudBridge logs to `/usr/share/cloud/bridge/logs/`.

10.2 CloudBridge can't connect to CloudStack

Check the file `/usr/share/cloud/bridge/conf/ec2-service.properties` and be sure that `managementServer` settings are correct. If you make any changes to this file, restart CloudBridge.

11 More Information

- [CloudStack API](#)
- [Amazon EC2 API](#)