WOWZA®
MEDIA SYSTEMS

Wowza Media Server® 2

Wowza Server for EC2 Edition

Wowza Media Server 2
for EC2 Edition

®

Version 2.0.0

http://www.wowzamedia.com

Third-Party Information

This document contains links to third-party websites that are not under the control of Wowza Media Systems, Inc. ("Wowza") and Wowza is not responsible for the content on any linked site. If you access a third-party website mentioned in this document, then you do so at your own risk. Wowza Servervides these links only as a convenience, and the inclusion of any link does not imply that Wowza endorses or accepts any responsibility for the content on third-party sites.

Trademarks

Wowza, Wowza Media Systems, Wowza Media Server and related logos are trademarks of Wowza Media Systems, Inc., and may be registered in the United States or in other jurisdictions including internationally.

Adobe and Flash are registered trademarks of Adobe Systems Incorporated, and may be registered in the United States or in other jurisdictions including internationally.

Other product names, logos, designs, titles, words, or phrases mentioned may be trademarks, service marks or trade names of other entities and may be registered in certain jurisdictions including internationally.

Third Party Copyright Notices

Log4j and Mina: Copyright © 2006 The Apache Software Foundation

Java ID3 Tag Library and JLayer 1.0 (classic): Copyright © 1991, 1999 Free Software Foundation, Inc.

Java Service Wrapper: Copyright © 1999, 2006 Tanuki Software, Inc.

Table of Contents

What's New

Change	Description	Release Date
Initial Release	Document release	
Update to doc v1.01	Update to include configuration instructions for use with Elasticfox	November 12, 2009
Update to doc v2.0.0	Wowza Media Server 2 Update	January 15, 2010
Doc v2.0.0.4	Information on monitoring with Cacti	February 3, 2010
Doc v2.0.0.5	Fixed <Download> doc bug	March 12, 2010
Doc v2.0.0.6	Fixed URL doc bug in Cacti section	April 15, 2010
Doc v2.0.0.7	Clarified port information and added Cacti information regarding elastic ip	October 7, 2010
Doc v2.0.0.8	Changed port range for RTSP/RTP UDP Streams	December 22, 2010

Introduction

Amazon Elastic Cloud Computing (Amazon EC2) is a web service from Amazon that provides flexible, easy to provision computing resources. Wowza Media Server 2 (Wowza Server) is a Java based, high-performance, extensible and fully interactive media streaming software platform that provides live and on-demand streaming, chat and remote recording capabilities to a wide variety of media player technologies. Wowza Server can deliver content to many popular media players such as Adobe's® Flash® Player, Microsoft's Silverlight® player, Apple's iPhone™ and iPod® touch and Apple's QuickTime® player, among others. Wowza Media Server 2 includes support for many streaming protocols including the Real-Time Messaging Protocol (RTMP), Microsoft Smooth Streaming, Apple HTTP Live Streaming, Real-Time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP), MPEG2 Transport Streams (MPEG-TS) and more. It is an alternative to the Adobe Flash Media Server products (FMIS and FMSS), Apple Streaming Server (Darwin) and other media servers. This document describes how to install and configure Wowza Server on Amazon EC2.

If you have not already signed up or are interested in learning more about Wowza Server on Amazon EC2, visit the following page on the Wowza Media Systems website:

http://www.wowzamedia.com/ec2.html

This document assumes you have signed up for Amazon EC2 and have installed the EC2 command line tools or installed and configured Elasticfox. You can obtain more information about Amazon EC2 and download the command line tools and documentation by going to:

http://www.amazon.com/ec2

Elasticfox is a developer tool that allows you to manage your Amazon EC2 account by using it as a Firefox plug-in. Many users will find Elasticfox easier than the command line tools to manage EC2 and the Wowza AMIs. This document will describe how to manage instances from both the command line as well as by using Elasticfox. If you intend to follow the directions for Elasticfox this document assumes that you have both Mozilla Firefox and Elasticfox installed on your

computer. Mozilla Firefox is available for download by going to:

http://www.mozilla.com/firefox

The Elasticfox software download, installation instructions and *Getting Started Guide* are available by going to:

http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609

Note: This document does not describe using the Amazon EC2 AWS Management Console because at this time the AWS Management Console does not support the loading of user files at startup. This functionality is needed to load Wowza Startup Packages which you'll learn about later in this document.

This document also assumes you are familiar with Wowza Server. You can download a free Wowza Server Developer edition license by going to:

http://www.wowzamedia.com/store.html

The Wowza Server download includes the Wowza Server software, documentation and examples. Once you have your client side and server side application up and running on your local machine, use this document to learn how to deploy it on Amazon EC2.

In Amazon EC2 a single virtual machine configuration is registered as an item called an AMI (Amazon Machine Image). You only pay for what you use with Wowza on EC2. You pay by the hour and for bandwidth used, instead of purchasing a license. You are billed directly by Amazon.

This document is meant to help users specifically with Wowza Server for EC2. Once you're familiar with the basic tutorials in this doc see the Wowza Media Server 2 for EC2 Quick Start Guide for additional examples. The Wowza Media Server 2 User's Guide contains comprehensive documentation for the Wowza Media Server.

Starting a Wowza Server Instance using Elasticfox

Elasticfox is a developer tool that allows you to manage your Amazon EC2 account from the Elasticfox Firefox plug-in. Many users will find Elasticfox easier than the command line tools to manage EC2 and the Wowza AMIs. If you prefer to use the command line tools to manage Wowza Server you should skip to the next chapter "Starting a Wowza Server instance using the command line tools". If you intend to follow the directions for Elasticfox this document assumes that you have both Mozilla Firefox and Elasticfox installed on your computer. Mozilla Firefox is available for download by going to:

http://www.mozilla.com/firefox

The Elasticfox software download, installation instructions and *Getting Started Guide* are available by going to:

http://developer.amazonwebservices.com/connect/entry.jspa?externalID=609

After signing up for Wowza Server for Amazon EC2 you will need to get the current Wowza Server DevPay AMI IDs. You can get these IDs from the Wowza Server EC2 Support page: http://www.wowzamedia.com/ec2support.html. These AMI IDs are what will be used to startup a Wowza Media Server 2 EC2 instance. For the following examples we will be using the AMI ID AMI-123456. Please substitute the AMI ID that you received from this page when starting your own instance.

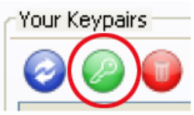
The Wowza Server AMI includes the Fedora Core 8 operating system, Java 6 JDK along with the Wowza Server. For a more detailed description of the environment, see the "Wowza Server Instance Details" chapter of this document.

Starting an Instance

Start Elasticfox by clicking on the 'Tools' menu in Firefox then selecting 'Elasticfox'. Elasticfox will ask you to provide your AWS Credentials, select 'Yes' at the first prompt then enter your AWS Account name, Access Key and Secret Access Key. You can create or view your account's key by logging in at the following URL: http://aws-portal.amazon.com/gp/aws/developer/account/index.html?action=access-key

Once you've entered your Account Credentials click the 'Close' button. Check your access by Clicking on the 'Availability Zones' tab in Elasticfox. You should see several regions available. If you do not see any regions then close and restart Firefox and Elasticfox. This is sometimes necessary the first time you enter new credentials.

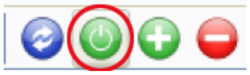
The next step is to set up KeyPairs for your instances. These are used to connect to your Wowza AMI instance for administration purposes outside of Elasticfox. Click on the 'KeyPairs' tab, then



on the Green, 'Create a KeyPair' button. Name your keypair (for example: wowza-keys) and then save the file somewhere safe. We'll be using this file later to telnet to the server so remember where you save it.

Next we'll start a small Wowza instance. Before you can start a Wowza instance you'll need to subscribe to Wowza Server for Amazon EC2 from your Amazon Web Services (AWS) account. To see information on pricing and to activate your subscription go to the following URL: https://aws-portal.amazon.com/gp/aws/user/subscription/index.html?offeringCode=F2CD62C1

Obtain the Wowza Server DevPay AMI ID for a small instance type from here (http://www.wowzamedia.com/ec2support.html). Click the 'Images' tab in Elasticfox then enter the AMI ID in the 'Images' text box. The long list should now have been reduced to a single line with the details of the current Wowza AMI. Click the AMI name to select it then click the 'Launch Instance' button.



The default values are set to launch a small instance with the keypair you just created. Click the 'Launch' button at the bottom of the window. Elasticfox will move you to the 'Instances' tab which lists all of your currently running instances as well as recently terminated ones. The 'State' will show 'pending' for up to several minutes as EC2 starts up your instance.

Note: You can start different size instances by specifying a different instance type using the 'Instance Type' drop down menu. The m1.small and c1.medium instance types uses the i386 (32-bit) image type and the m1.large, m1.xlarge and c1.xlarge instance types use the x86_64 (64-bit) image type. Wowza Server provides prebuilt images for both instance types. The following documents describe the different instance types:
http://aws.amazon.com/ec2/instance-types/
http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?CLTRG-run-instances.html

Based on our customer's results we recommend the following guidelines for sizing and bandwidth of instance size:

Small: 150Mbps
Large: 250Mbps
Extra Large: 350Mbps

Once the 'State' of your instance shows 'running' we are ready to test.

Testing the instance

You can quickly test your running Wowza Server instance using the **SimpleVideoStreaming** example that ships with Wowza Server (you will need to install Wowza Server on your local machine or download a set of example files from here). First open up port 1935 for RTMP streaming.

Note: Use caution when opening ports on your server. Reference the Amazon security guide here for best practices when using an Amazon AMI:

http://aws.amazon.com/security/

Amazon EC2 has a concept of 'Security Groups' which manage access to EC2 machines. You can view and change these security settings from Elasticfox on the 'Security Groups' tab. Make sure the 'default' group is selected in the left 'Your Groups' pane, then click the 'Grant Permission' button on the right side 'Group Permissions' pane of your browser window. Choose 'HTTP' in the 'Protocol Details' dropdown, 'TCP/IP' as the Protocol and '1935' as the Port. Leave the 'Host' radio button selected and in the field next to it enter 0.0.0.0/0. Port 1935 will now be open in any new AMIs you launch in the 'default' security group.

To see the public address of your server in the Elasticfox interface just double-click the instance name and view the 'Public DNS Name' entry.

Stream the example video

Now that port -1935 is open and Wowza Server is installed on your local machine double click:

```
[install-dir]/examples/SimpleVideoStreaming/client/simplevideostreaming.html
```

Enter the following information:

```
Server: rtmp://[instance-public-domain]/vod
Stream: merry_melodies_falling_hare
```

where [instance-public-domain] is the public domain name for the instance.

Click "Play". It should start playing the "Merry Melodies" sample video.

Opening a Telnet Session to an Instance

Before connecting to your Wowza Server EC2 instance using telnet over SSH, you must open up TCP port 22. To do this, select the 'Security Groups' tab, confirm that the 'default' group is selected in the left pane then click the 'Grant permission' button in the right pane. In the dialog box that appears select 'SSH' from the 'Protocol Details' drop down and 'TCP/IP' as the 'Protocol'. Leave the port as '22'. In the 'Host/Network Details' section enter 0.0.0.0/0 as 'Host' then click the 'Add' button. This will allow any machine on the internet to connect to port 22.

Putty uses a private security key to authenticate users instead of passwords. In order to get a telnet session started we'll have to first convert the Amazon EC2 key into Putty format. First, download putty.exe and puttygen.exe from the "Downloads" section of the Putty website:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

Run the puttygen.exe tool, click the "Load" button and select the .pem key file that you created and saved earlier, click the "Save private key" button and save the key somewhere safe with the filename your-keypair-putty.ppk. Puttygen will ask if you want to protect your key with a passphrase. A passphrase will give an additional layer of security by requiring any connections to the server to have the proper security key as well as type in the passphrase. Adding a passphrase is not a requirement when creating a putty key.

In Elasticfox in the upper right corner there is a 'Tools' button. By clicking this we can configure Elasticfox to launch putty automatically for us. After clicking 'Tools' the first text field is labeled 'SSH Command:' Place the full path to putty.exe on this line (for example: "C:\aws\tools\putty.exe") then click the 'OK' button.

In the Elasticfox interface 'right-click' on the running instance you want to connect to and select 'Connect to Public DNS Name'. Elasticfox will ask you to Select the EC2 Private Key File for key: *yourkeyname*. Choose the .ppk file you created and click 'OK'. You'll then be asked if you want to use that key file as the default EC2 Private Key File for the account you're currently logged on as. Either click OK or Cancel and then putty will launch a terminal console window. Because you used keypairs as your authentication method it will automatically log you in as **root** unless you used a passphrase for your .ppk key in which case you'll be prompted to enter that.

The Wowza Server application and its supporting files can be found under symbolic links at /home/Wowza. Under this path you'll find the application, keys, logs, etc…For more information see Wowza Server Instance Details later in this doc or the *Wowza Media Server User's Guide* at: http://www.wowzamedia.com/resources/WowzaMediaServer_UsersGuide.pdf

You can end your telnet session by entering the "exit" command.

Telnet command cheat sheet for Linux beginners

Connecting to an instance with a Telnet session makes you a linux user. Congratulations! If that is not a familiar environment for you here are some simple commands you may find useful. If the command you're looking for is not on this list there are many guides posted on the internet regarding Linux.

```
cd /home/wowza
```

This will change your directory to one that shows the most common user accessed Wowza Media Server Directories. From here you can get a list of the 'wowza' sub-directories by typing: `ls` This will list the files and directories in the 'wowza' directory. To change directories simply type `cd [directory name]` to move into a directory and to move back up a level type `cd ..`

If you want to see the last part of one of the log files navigate into the log directory and type

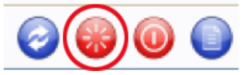
`tail -n 50 [logfilename]` – remember just type `ls` once you get to the logs directory and you'll see all the log file names. The `tail` command will display the last 50 lines of the log file. Handy to check for errors or connection status. Add a `-f` modifier and the display will update in real time. Press `[Control-C]` when you are ready to return to the command prompt.

Finally, if you need to stop or start the Wowza Media Server without rebooting the AMI use the command `service WowzaMediaServer stop` and `service WowzaMediaServer start`.

Stopping an Instance

When you stop an instance you will lose all changes or files you have on the server. If you have anything you don't want to lose save it to Amazon S3 or to a local machine before stopping the instance. **Be aware that Amazon recommends you confirm that the machine reaches a status of 'terminated' as they will continue to charge for instances that fail to shut down correctly.**

On the 'Instances' tab of Elasticfox select the instance you want to stop running and click the 'Terminate Selected instances' button. The 'State' info will show 'shutting-down' and finally 'terminated'.



Starting a Wowza Server Instance using the command line tools

This document describes two ways in which to manage a Wowza Server Instance: from Elasticfox and from the Command Line. This section describes using the command line tools. If you followed the previous section to set up Wowza Server using Elasticfox you should skip to the next chapter "Configuring Wowza Server".

After signing up for Wowza Server Unlimited for Amazon EC2 you will need to get the current Wowza Server DevPay AMI IDs. You can get these IDs from the Wowza Server EC2 Support page: http://www.wowzamedia.com/ec2support.html. These AMI IDs are what will be used to startup a Wowza Media Server 2 EC2 instance. For the following examples we will be using the AMI ID AMI-123456. Please substitute the AMI ID that you received from this page when starting your own instance.

In this section, where applicable, all commands will include the --region command line parameter to specify the region in which the command is to be run. The "us-east-1" will be used for all example commands. Amazon EC2 currently supports 4 regions "us-east-1" (US), "us-west-1" (US), "ap-southeast-1" (Asia Pacific) and "eu-west-1" (Europe).

The Wowza Server AMI includes the Fedora Core 8 operating system, Java 6 JDK along with the Wowza Server. For a more detailed description of the environment, see the "Wowza Server Instance Details" chapter of this document.

Installing and configuring the command line tools

The following guide from Amazon describes in detail the procedure to install the command line tools, start an instance and connect to it through a telnet session. If you are new to Amazon EC2 it is probably important that you read through this document:

http://docs.amazonwebservices.com/AWSEC2/2008-12-01/GettingStartedGuide/

The following procedure will describe how to start an instance, connect to it using the putty telnet client and how to stop your instance. If you have not done so already follow the instructions in the Getting Started Guide above to install the most recent Java JDK (http://java.sun.com/javase/downloads), the API command line tools and configure the tools for your account.

See this forum post for instructions on how to setup the EC2 command line tools:

http://www.wowzamedia.com/forums/showthread.php?t=5553

Starting an Instance

These instructions assume you have the command line tools properly configured and that the following command lists the publicly available AMI IDs:

```
ec2-describe-images --region us-east-1 -x all
```

So let's get started by starting our first EC2 instance. First, download putty.exe and puttygen.exe from the "Downloads" section of the Putty website:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

Next, we are going to generate a key pair to use to launch and connect to our Wowza Server EC2 instance. Use the following command to generate a key pair:

```
ec2-add-keypair --region us-east-1 wowza-keypair
```

The output of this command will look something like this:

```
KEYPAIR wowza-keypair      6c:47:53:ac:4c:d5:87:8f:9d:5e:43:54:27:54:8f:e6:f9:1e:6c:92
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAkf63aqFW3oSY3fh4MkTE3FSJecm/p+pLHoXBmOLYhlHF8LDwu8PgO4z6Qqx/
jCwZqiqW6lZJgA93KLr0yHc1+4maaE83Nitt+v1lny8+XGFbh2khVTBrGDqzXLKIkERmx/imcXma
jdLJ6vo1NOWHBeyF+TWBXlSzyTmXGaPBHHIKfuN5bPXDFh5B/nFsbhLHmo5oINptVmy/eVYy65MY
olDIr8kjAn1Z2vSWbFuEYJAmWpFRBrlMfzt0I34+sfws01o1Wl+0cN+6b31trXHusUrEKqPbpU4r
XW+VuSt8yC2x+ByAr7fvuqEKNP8HEsCejZNmw+yHpTzBisAIDNwkSwIDAQABAoIBAEcNyQbEbxrt
PdkKTLVqcsUo/5+eaQC4Lsc0IfzeqrL0qXlzJDa+JivVeRdqDMmTzQncoIZ8aFlDLaNNGkVWf5dk
aRnLM/vRNz2ohgTrYVZkq79Dk016uySXrSO4YG6WW8+HXLptjKLprAwIqkyMSSLA8pNPTx8x6i5N
MRTxddL5kFit2bnXsd1xmWDw4/FQtHvqz6gkhjylgkLdmX8+K/spw5O2ceInoDax3ml6/kpqmgUG
Zie/q9c9BSicVkCS4ol3e5TRKzJ6TV5cjXRT59nexVYMxdcm7FuHI0b+BWMS4R3UIp5+t2nkUJFG
5H2TLUsXcMgC8PFWYF9Rz4o7KIECgYEA2Vym3R5nn73sp9h6gpQ4lMLLXH6ci4oTIpd4p7m+VVQT
YuI7nST/32CTpzP0OGlj2nSybAR7U0RrJv0xcQaQNGbbfaRRdhlkbz+Iwim8TApwNLDBSRKa3rTP
1XGbf8IZCGpu1PEkgKscbvUnTotdk1hfBsZlTHFWcOZkhmcb5VkCgYEAq/Jqb68tidgytAEGu9wg
ZXdj7HgF57Iv4s1yusxywS0P4kJ16dVS0ynokP/TCm3jQW9WjdQHfuvEEQFNxpE3Kni+HYL4u8a2
C75rXHdvSxbH0WE2RPo7EpgoluFEgg4rOuqL1E2fQ9fiI4T/7vEkKRQp0Eu0XL/hSwKZJ+peTkMC
gYEAoGOMwWcZIbd8dFH5/Haw1ZxzgPnwO8sT2ZeZsXIO71nNwE6dnJRbtpHGAkAE+sdRpjYsHiak
dcnro2Z7YCJusCwh1DE25wVPpgNkv5kaV6XwLAEtbuUqoCEdpytgicJIxHv88mZKACPDJGEDsyKF
Vx+DZzVoNCMBnOd0z/QvkmkCgYBxdHvsg48ttD0IbuDtlfq+WL/TKQLkdbuJEHzSatKRSlV/utCo
4PxMnwt4mYS8DfP0mi1nWe80PQbbHZwfGHRpDtCLEn9jD2PRbezTT29dAYQJUPaJHv9ZbpLhs3QL
yKOXtWlcRXeAQXNScpCZLf3pKQfKDBnPVUCnyOgb4VYmlwKBgHLk/iL9g/hkL+zz8o0p+SkzWYDH
CY1Gg97juvPbEEAmbrEhM9wZPno9cCrlVWRHf5S9idsmbW+BVOWIv0x+qxdEpbpLdtGBY+g8m6nD
LAx4pHJiUoYrt8fc4uq2zIYx1sPeaE1FppPCeahjWsYt1d9QYJQHTJkRnH3jJ+edAzv+
-----END RSA PRIVATE KEY-----
```

Copy the output from the command (excluding the first KEYPAIR line) and save it in a new text file named “wowza-keypair.pem”. Be careful not to include any extra line breaks in the new text file as puttygen will not successfully load the .pem file if it finds any. Run the puttygen.exe tool, click the “Load” button and select the newly saved file, click the “Save private key” button and save the key to the filename “wowza-keypair-putty.ppk”. This is the file you will use when using putty.exe to telnet to your running instance.

Before you can start a Wowza instance you'll need to subscribe to Wowza Server for Amazon EC2 from your Amazon Web Services (AWS) account. To see information on pricing and to activate your subscription go to the following URL: https://aws-portal.amazon.com/gp/aws/user/subscription/index.html?offeringCode=F2CD62C1

Next, we are going to start a small Wowza Server EC2 instance. Obtain the Wowza Server DevPay AMI ID for a small instance type from here (http://www.wowzamedia.com/ec2support.html). For this example we will use the AMI ID “ami-123456”. Use the following command to start your instance:

```
ec2-run-instances ami-123456 -k wowza-keypair --region us-east-1 -t m1.small
```

You can start different size instances by specifying a different instance type using the –t command line option. The m1.small and c1.medium instance types uses the i386 (32-bit) image type and the m1.large, m1.xlarge and c1.xlarge instance types use the x86_64 (64-bit) image type. Wowza Server provides prebuilt images for both instance types. The following documents describe the different instance types:

http://aws.amazon.com/ec2/instance-types/

http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide/index.html?CLTRG-run-instances.html

To start an extra large instance, execute the command (where "ami-654321" is the current x86_64 AMI ID found on the Wowza Server EC2 Support page: http://www.wowzamedia.com/ec2support.html):

```
ec2-run-instances ami-654321 -k --region us-east-1 wowza-keypair -t m1.xlarge
```

It will take several minutes for the AMI to start. You can check on the progress by executing the command:

```
ec2-describe-instances --region us-east-1
```

This will return information on all instances that have been started or stopped by you. Once your instance is up and running, the ec2-describe-instances command will return the following output (notice the status value running):

```
RESERVATION  r-49bb5320  664010435657  default
INSTANCE     i-1e3dcb77  ami-3dda3f54  ec2.amazonaws.com domU.internal running
air     0    8ED157F9    m1.small      2007-12-12T22:04:44+0000
```

The i-1e3dcb77 value is the instance id. This id will be used to stop a running instance. The first domain listed (ec2.amazonaws.com) is the public domain (or ip address) assigned to this instance. The second domain (domU.internal) is the internal domain (or ip address) assigned to the instance..

Testing the instance

You can quickly test your running Wowza Server instance using the **SimpleVideoStreaming** example that ships with Wowza Server (you will need to install Wowza Server on your local machine). First open up port 1935 for RTMP streaming.

Note: Use caution when opening ports on your server. Reference the Amazon security reference here for best practices when using an Amazon AMI.

http://aws.amazon.com/security/

Execute the following Amazon EC2 command on your local machine (the same console windows in which you started the EC2 instance):

```
ec2-authorize default --region us-east-1 -p 1935
```

Stream the example video

Now that port -1935 is open and Wowza Server is installed on your local machine double click:

```
[install-dir]/examples/SimpleVideoStreaming/client/ simplevideostreaming.html
```

Enter the following information:

```
Server: rtmp://[instance-public-domain]/vod
Stream: merry_melodies_falling_hare
```

where [instance-public-domain] is the public domain name for the instance.

Click "Play". It should start playing the "Merry Melodies" sample video.

Opening a Telnet Session to an Instance

Before connecting to your Wowza Server EC2 instance using telnet over SSH, you must open up TCP port 22. To do this, execute the following command:

```
ec2-authorize default --region us-east-1 -p 22
```

To connect or telnet to your running instance, follow these steps:

1. Startup the putty.exe telnet client and enter the public domain name for the instance returned by the "ec2-describe-instances" command into the "Host Name (or IP address)" field and select the "SSH" protocol radio button.

2. Navigate to the "Connection->SSH->Auth" settings panel in the navigation menu on the left, select the "Browse…" button next to the "Private key file for authentication" field and open the "wowza-keypair-putty.ppk" file generated using puttygen.exe above.

3. Click the "Open" button in the bottom right of the putty.exe window. It should connect to your instance and present you with a security warning dialog. Click the "Yes" button in this dialog.

4. You should now be prompted to enter a login name. Enter the username "root" followed by the enter key. You are now logged into your instance as the root user.

You can end your telnet session by entering the "exit" command.

Stopping an Instance

When you stop an instance you will lose all changes or files you have on the server. If you have anything you don't want to lose save it to Amazon S3 or to a local machine before stopping the instance. **Be aware that Amazon recommends you confirm that the machine reaches a status of 'terminated' as they will continue to charge for instances that fail to shut down correctly.**

To stop a running instance, execute the command:

```
ec2-terminate-instances --region us-east-1 i-[instance-id]
```

Where [instance-id] is the id of the instance returned by ec2-describe-instances. Again, this process can take a few minutes. You can check the shutdown status by executing the command:

```
ec2-describe-instances --region us-east-1
```

Wowza Server Configuration

The method described above starts a generic Wowza Server instance with most of the example applications installed. You can configure a Wowza Server instance at startup time by passing in user data in the form of a startup package. A startup package is a zip archive of a folder that contains a startup manifest (startup.xml) along with configuration files and scripts. Amazon EC2 limits the size of a valid startup package to 16kB.

Startup Package Example

Below is the file structure of a simple startup package:

[mywowzaconfig]

```
  startup.xml
  [wowza]
    [applications]
      [myapp]
    [conf]
      [myapp]
        Application.xml
  [tuning]
    [bin]
      m1.large-setenv.sh
      m1.small-setenv.sh
      m1.xlarge-setenv.sh
      setenv.sh
    tuning.sh
```

Here are the contents of startup manifest (startup.xml) for the startup package outlined above:

```
<Startup>
   <Commands>
      <Install>
         <Package>com/wowza/wms/WowzaMediaServerPro1.3.2-patch1.zip</Package>
      </Install>
      <Install>
         <Folder>wowza</Folder>
      </Install>
      <RunScript>
         <Script>tuning/tune.sh</Script>
      </RunScript>
   </Commands>
</Startup>
```

Using a Startup Package

From Elasticfox

When you Launch a new instance using Elasticfox you can load a Startup Package by clicking the 'Open Binary File' button in the 'Launch new instance' dialog and selecting your startup package zip file.

From the command line

The startup package is sent to Amazon EC2 as part of the ec2-run-instances command. Once you have created and organized your startup package, use a zip utility to zip up the startup folder (in this case [mywowzaconfig]) into a single zip file. The zipped up startup package cannot exceed 16kB (kilo-bytes) in size (see the <Download> command as described in the "Startup Package Reference" chapter of this document for suggestions on how to work around this limit). Next, when you startup the Wowza Server AMI use the following command:

```
ec2-run-instances ami-123456 --region us-east-1 -t m1.xlarge -f [path-to-startup-pkg]mywowzaconfig.zip
```

where [path-to-startup-pkg] is the full directory path to the startup package.

This will send the startup package as user-data to the Wowza Server AMI. When the Wowza Server instance is started it will check to see if it was sent user-data. If it was, it will save the user data to a file on the file system, unzip the file, locate the startup manifest and execute the commands in the startup manifest. The startup package details are covered in the "Startup Package Reference" chapter of this document.

Startup Package Basics

Let's take a quick look at the example startup package. The first command:

```
<Install>
   <Package>com/wowza/wms/WowzaMediaServerPro1.3.2-patch1.zip</Package>
</Install>
```

Instructs the startup processor to install patch WowzaMediaServerPro1.3.2-patch1.

The second command:

```
<Install>
  <Folder>wowza</Folder>
</Install>
```

Instructs the startup processor to copy the contents of the included [wowza] folder into the /usr/local/WowzaMediaServerPro folder on the running Wowza Server instance. This gives you a chance to create application folders and configuration folders and files.

The last command:

```
<RunScript>
  <Script>tuning/tune.sh</Script>
</RunScript>
```

Instructs the startup processor to run the included script tuning/tune.sh. This script might change configuration parameters based on instance size; small, large or xlarge. This script must be a shell script (not a binary application) and will be executed by the operating environment that is running on the Wowza Server instance. When a script is executed the working directory is set to the root directory of the startup package (the folder that contains the startup.xml file).

Pre-built Startup Packages

Wowza Media Systems provides several pre-built startup packages that can either be used as is or modified to suite your needs. You can download these packages from the following web site:

http://wowzamediasystems.s3.amazonaws.com/packagelist.html

Wowza Server Instance Details

A Wowza Server Amazon EC2 AMI is the base Amazon Fedora Core 8 instance (ami-2b5fba42 i386 and ami-2a5fba43 x86_64) with the following additional items installed; Java 6 Development Kit, vsftpd (FTP server) and Wowza Media Server for EC2.

Wowza Server Details

Wowza Server is installed at its default location:

/usr/local/WowzaMediaServer

The Wowza Media Server service is running on the following ports:

```
1935   - rtmp port
80     - rtmpt port
443    - rtmps port

The Wowza Media Server is managed using the following ports :

8084   - JMX/JConsole Management
8085   - JMX/JConsole Management
21     - FTP access
22     - SSH access
```

These ports will need to be opened for the Wowza Server instance to be available on the Internet. The commands are as follows:

```
ec2-authorize default --region us-east-1 -p 1935
ec2-authorize default --region us-east-1 -p 80
ec2-authorize default --region us-east-1 -p 443
```

All ports are configured to return load balancing information over http as outlined in the "Scalability for Video On Demand Applications" section of the Wowza Media Server User's Guide. What this means is if you open a web browser and enter the url:

```
http://[instance-public-domain]:1935
```

where [instance-public-domain] is the public domain name for the instance.

Wowza Server will return load information for the instance. Example output is:

```
server=864
```

Where 864 is the current number of Flash client connections to the instance.

If you wish to do RTSP/RTP based stream then you will need to open up UDP ports 1-9999 to UDP traffic. This is done with the following command:

```
ec2-authorize default --region us-east-1 -P udp -p 6970-9999
```

FTP Access

The Wowza Server Amazon instance comes pre-installed with the "vsftpd" FTP server. A user named "wowza" has been added to the system with the password "password". This account can be used to upload content or configure the server using ftp. Be sure you open up the proper ports for FTP and SSH:

```
ec2-authorize default --region us-east-1 -p 21
ec2-authorize default --region us-east-1 -p 22
```

Also, you will need to setup your ftp client to use the PORT (also known as ACTIVE) communication method (rather than the PASV). Consult your ftp client's documentation for more information.

The following Wowza Server folders have been relocated to the home directory of user "wowza" ("/home/wowza").

```
[install-dir]/applications    →  /home/applications
[install-dir]/conf            →  /home/conf
[install-dir]/lib             →  /home/lib
[install-dir]/logs            →  /home/logs
[install-dir]/content         →  /home/content
```

Symbolic links have been created to their default locations. The "/home/wowza" folder also contains the folder "content" that can be used to store .flv content for streaming. The following sample video is available in this folder: "merry_melodies_falling_hare.flv". The Streams/StorageDir setting in Applications.xml has been configured to point to this content folder.

This additional configuration is done to make it easier to configure the Wowza Server using ftp. When you log into a Wowza Server Amazon instance as the user "wowza" (default password "password") through ftp, the base folder will be "/home/wowza". From there you can upload content, create application folders and upload and change configuration folders and files.

You can easily change the password for the default "wowza" user by logging into the instance as the "root" user and executing the following command (follow the prompts):

```
passwd wowza
```

For Security reasons we strongly recommend that you change the default password for the "wowza" FTP account for your AMI.

Java Management Extension (JMX)

The JMX/JConsole interface (as described in the "Server Management Console and Monitoring" chapter of the Wowza Media Server User's Guide) to your instance is preconfigured to listen to connections on TCP ports 8084 and 8085 using the public domain name. You will need to open these ports to TCP traffic to be able to successfully connect to your EC2 instance.

```
ec2-authorize default --region us-east-1 -p 8084
ec2-authorize default --region us-east-1 -p 8085
```

The JMX url is:

```
service:jmx:rmi://[public-domain-name]:8084/jndi/rmi://[public-domain-name]:8085/jmxrmi
```

From most JMX tools such as JConsole you should be able to connect using the address:

```
[public-domain-name]:8085
```

Where [public-domain-name] is the public domain name of the instance. The default username is "admin" and password is "admin". User access is managed in the following two files and is described in the Wowza Server User's Guide:

```
/usr/local/WowzaMediaServerPro/conf/jmxremote.access
/usr/local/WowzaMediaServerPro/conf/jmxremote.password
```

Default Startup Package

If you start the Wowza Server EC2 AMI without specifying a startup package the default startup package will be used. If you want to take a look at the default startup package it can be downloaded at:

http://wowzamediasystems.s3.amazonaws.com/com/wowza/startup/default.zip

The default startup package includes all the configuration and application files needed to run the following list of the examples that are included with Wowza Media Server:

FastPlayVideoStreaming

LiveVideoSteraming

LoadBalancer

MediaSecurity

NativeRTPVideoStreaming

RemoteSharedObjects

ServerSideModules

SHOUTcast

SimpleVideoStreaming

TextChat

VideoChat

If you supply your own startup package by default their will not be any applications configured in the applications and conf folders. Your startup package needs to provide this configuration.

Custom Module Development

There are several System level properties that are available when developing custom server side modules. These properties describe the current running instance. You can get the value of one of these system properties by executing the Java method:

```
String value = System.getProperty(“com.wowza.amazonaws.ec2.AWSEC2_METADATA_INSTANCE_ID”);
```

The available properties are:

```
com.wowza.amazonaws.ec2.AWSEC2_METADATA_INSTANCE_ID         - Amazon instance id
com.wowza.amazonaws.ec2.AWSEC2_METADATA_SECURITY_GROUPS     - Security group
com.wowza.amazonaws.ec2.AWSEC2_METADATA_LOCAL_IPV4          - Local IP address
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_LAUNCH_INDEX    - Launch index
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_HOSTNAME     - Public host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PRODUCT_CODES       - DevPay product code
com.wowza.amazonaws.ec2.AWSEC2_METADATA_INSTANCE_TYPE       - instance type (m1-small,
                                                              m1-large, m1-xlarge)
com.wowza.amazonaws.ec2.AWSEC2_METADATA_HOSTNAME            - Public host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_LOCAL_HOSTNAME      - Local host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_IPV4         - Public IP address
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_MANIFEST_PATH   - S3 manifest path
com.wowza.amazonaws.ec2.AWSEC2_METADATA_RESERVATION_ID      - Instance reservation ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_ID              - AMI ID
```

Streaming Media Directly from S3

You can use Wowza Server EC2 to stream media directly from the Amazon Simple Storage Service (S3). Wowza Server employs a read through local disk cache to improve the performance of streaming media from S3. The configuration of the caching mechanism is done automatically and is adjusted based on the AMI instance type being used (m1-small, m1-large, m1-xlarge). This feature requires that you use “Wowza Server DevPay AMI ID v EC2 1.1.10” or greater.

This feature is included in the default startup package for Wowza Server EC2 edition.

Where “ami-123456” is the current Wowza Server AMI id. To stream content, use stream names in the form:

```
[media-type]:amazons3/[s3-bucket-name]/[path-to-content-in-s3]
```

For example to play the file “mycoolvideo.m4v” that is stored in S3 bucket “mybucket” at the path “videos/coolvideos” from the Amazon instance “ec2-75-101-208-8.compute-1.amazonaws.com”, the connection information is:

```
Server: rtmp://ec2-75-101-208-8.compute-1.amazonaws.com/vods3
Stream: mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v
```

By default the vods3 feature is configured with S3 authorization turned off. This means that all content must be publicly available. To stream non-publicly available content, unzip the default.zip startup package and modify the two system properties “awsAccessKeyId” and “awsSecretAccessKey” in “wowza/conf/Server.xml”. Set these two values to the “Access Key

ID" and "Secret Access Key" for the user in which you would like to authorize access and zip up the default folder. Use this new zip archive as your startup package. For example:

```
<Property>
  <Name>awsAccessKeyId</Name>
  <Value>11DV8PNKTHN1234732</Value>
</Property>
<Property>
  <Name>awsSecretAccessKey</Name>
  <Value>p0fsdFIE1Nofyx5Sfe+CmuQi0uXt7ygrD8Xxz+</Value>
</Property>
```

The default setup is such that any content in S3 that is publicly available can be re-streamed through your instance. To limit the content to specific buckets you can use the stream alias system to add aliases for the buckets from which you would like to stream. The stream alias package has been included with the default.zip startup package. To modify the default configuration edit the file "wowza/conf/aliasmap.play.txt" that is included in the default.zip startup package and create a new startup package with the modifications. To do this, unzip the default.zip startup package, add wildcard entries for the buckets from which you would like to stream content and comment out the default stream alias rule. For example, to limit streaming to content only coming from "mybucket" the "wowza/conf/aliasmap.play.txt" file should look like this:

```
mybucket/*=amazons3/mybucket/${Wildcard.Match1}
flv:mybucket/*=flv:amazons3/mybucket/${Wildcard.Match1}
mp3:mybucket/*=mp3:amazons3/mybucket/${Wildcard.Match1}
mp4:mybucket/*=mp4:amazons3/mybucket/${Wildcard.Match1}
# commented out *=${Stream.Name}
```

With this in place to play the file "mycoolvideo.m4v" that is stored in S3 bucket "mybucket" at the path "videos/coolvideos" from the Amazon instance "ec2-75-101-208-8.compute-1.amazonaws.com", the connection information is:

```
Server: rtmp://ec2-75-101-208-8.compute-1.amazonaws.com/vods3
Stream: mp4:mybucket/videos/coolvideos/mycoolvideo.m4v
```

The single url representation of this same content is:

```
rtmp://ec2-75-101-208-8.compute-1.amazonaws.com/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v
```

There are three sample files that you may use to test the system. The stream names are:

```
amazons3/wowzamediacache/sample/Extremists.flv
mp4:amazons3/wowzamediacache/sample/Extremists.m4v
mp3:amazons3/wowzamediacache/sample/Extremists.mp3
```

Monitoring the Wowza Media Server on EC2 using Cacti

Introduction

Cacti is a complete open source network graphing solution offering graph templating to simplify the representation of data gathered about the performance of the Wowza Server as well as the health of the AMI's OS. Cacti is included in the Wowza AMIs but is turned off by default. For detailed information about Cacti you can read more at: http://www.cacti.net/

Once activated on the Wowza Server it will begin to monitor and track information, constantly collecting and storing data about the server and its connections every 60 seconds. It stores collected data on the local AMI for up to 1 year. If you shut down your AMI you will lose the collected data and will start with a clean slate next time you start Cacti. You may add custom graphs to collect information for a specific VHost, Application, application Instance and or Stream. There are two accounts configured in the AMI.

```
User: guest Password: password
User: admin Pasword: password
```

The user 'guest' can only view existing charts. The user 'admin' has the ability to turn Cacti data gathering on/off as well as add/remove graphs.

The charts included in the AMI are broken into three groups:

```
Wowza Media Server
   Localhost – Wowza Server - Server Connections
   Localhost – Wowza Server - Application (live) Connections
   Localhost – Wowza Server - Application (rtplive) Connections
   Localhost – Wowza Server - Application (vod) Connections
   Localhost – Wowza Server - Application (vods3) Connections

Network
   Localhost – Traffic – eth0
   Localhost – Traffic – lo

System
   Localhost – CPU Utilization
   Localhost – Disk Space - /dev/sda1
   Localhost – Disk Space - /dev/sda2
   Localhost – Load Average
   Localhost – Logged in Users
   Localhost – Memory Usage
   Localhost – Processes
```

Getting started with Cacti

1) Open TCP port 8080 (instructions for opening ports are above in the 'Testing your Instance' section.

2) GoTo URL: http://[public-domain-name]:8080/cacti

Note: If you are using an Elastic IP you MUST restart the Wowza service after assigning the Elastic IP for Cacti and JMX to work correctly.

3) Login as:

```
User Name: admin
Password:  password
```

4) In the left hand column under the 'Management' section click the link for 'Devices'

5) On the 'Management: Devices" page click the 'Localhost' device in the 'Description' column.

6) Uncheck the 'Disable Host' setting in the 'General Host Options' section of the page. Then click the 'Save' button at the bottom of the page.

Your server is now collecting data and can be viewed on the 'Graph' tab. Because Cacti only collects data once every 60 seconds it will take a few minutes before your data will appear in the graphs.

You can change the time scale of your graphs as well as the displayed time period at the top of the 'graphs' page. The default Wowza Media Server graphs display connections for iPhone, Flash, RTSP and Silverlight over time with a Current/Average/Maximum number of connections displayed for each type. Use the left hand navigation column to switch between the graphs for the Network, System and Wowza Server graphs.

Adding additional Wowza Server graphs

You can add additional graphs to collect data at any level of the Wowza Server streaming hierarchy: VHosts, Application, Application Instance and Streams. For example, to add a graph that charts a particular live stream:

1) Login as admin

2) Navigate to the 'Management: Devices' page and select 'Localhost' device.

3) Click 'Create Graphs for this Host' link (in the upper right of the page)

4) Select 'Wowza Server - Stream Connections' from the 'Graph Templates: Create:' popup at the bottom of the 'Graph Templates' section and click the 'Create' button at the bottom of the page.

5) In the 'Title' and 'Name' fields near the top of the form change '([name])' to '(yourStreamName)'.

6) Most of the graph specific fields such as 'JMX URL', 'JMX Username', 'JMX Password' and 'Virtual Host Name' are filled in for you. Enter the 'Application Name' and 'Stream Name' for the stream you wish to monitor and adjust any of the other fields as needed.

7) Click the 'Create' button at the bottom of the window.

8) Click 'Management: Graph Management' in the left hand pane.

9) Select the check box for your new Graph in the 'Graph Title' pane.

10) In the 'Choose an action' pop-up select 'Place on a Tree (Default Tree)' then click the 'Go' button.

11) Choose where you would on the Graph tree you'd like your graph to appear then click the 'yes' button.

12) When you navigate back to the 'graphs' page your new graph will be there. It may take 5-10 minutes before the graph displays data.

Note: The stream name to use is the stream name that is used to publish or play the stream but without any prefixes such as mp4: or mp3:.

Note: If you are monitoring a multi-bitrate stream that is targeted to the iPhone or Silverlight use the .smil file name. For example if the iPhone URL is:

http://[wowza-ip-address]:1935/vod/smil:myvideo.smil/playlist.m3u8

then the stream name to use is "myvideo.smil".

If you are monitoring a multi-bitrate stream that is targeting Flash then you must specify the individual stream names in the multi-bitrate group separated by the "|" (pipe) character. For example, if the three stream names are "mystream_250kbps", "mystream_500kbps" and "mystream_750kbps", then specify the stream name: "mystream_250kbps|mystream_500kbps|mystream_750kbps".

To add a graph for a specific app, appInstance, VHost or Server in step Four above choose the corresponding graph type from the popup then complete the steps, substituting your specific information where needed.

Startup Package Reference

This section describes in detail each of the command that can appear in a startup manifest file (startup.xml). The three commands are; <Install>, <Download>, <RunScript>.

Command <Install>

The <Install> command will copy the contents of a folder into the Wowza Server installation folder. The <Install> command can either contain a single <Package> element or single <Folder> element.

```
<Install>
  <Package>[path-to-package]</Package>
</Install>
```

```
<Install>
  <Folder>[relative-or-absolute-directory-path]</Folder>
</Install>
```

Element <Install>/<Package>

A <Package> is a Wowza Server patch or set of configuration files provided by Wowza Media Systems. A list of all available packages can be obtained by going to the url:

http://wowzamediasystems.s3.amazonaws.com/packagelist.html

The packagelist.html file includes the package path that is needed to locate the package. Here is an example of an <Install> command that installs WowzaMediaServerPro-1.3.2-patch1:

```
<Install>
  <Package>com/wowza/wms/WowzaMediaServerPro1.3.2-patch1.zip</Package>
</Install>
```

You can download a package to your local machine by pre-pending "http://wowzamediasystems.s3.amazonaws.com/" to the package path.

Element <Install>/<Folder>

If you specify the <Folder> element then the contents of the specified folder will be copied into the Wowza Server installation folder. The folder directory can be either a relative or absolute directory path on the running Amazon instance. The base directory when calculating a relative file path, is the root directory of the startup package (the folder that contains the startup.xml file).

For example to install the contents of the folder wowza that is contained within the startup package, the <Install> command is:

```
<Install>
  <Folder>wowza</Folder>
</Install>
```

To install the contents of the folder /opt/wowza, the command is:

```
<Install>
  <Folder>/opt/wowza</Folder>
</Install>
```

Command <Download>

The <Download> command will download content from a web server and save it to the local Amazon instance. The <Download> command includes the following elements:

```
<Download>
  <URL>[URL]</URL>
  <Data>[data]</v>
  <Header><Name>[key-name]</Name><Value>[value]</Value></Header>
  <Header><Name>[key-name]</Name><Value>[value]</Value></Header>
  <Destination>[relative-or-absolute-file-path]</Destination>
  <Action>[UNZIP, INSTALL]</Action>
</Download>
```

The only two required elements are <URL> and <Destination>. To download a file from the url http://www.mycompany.com/myfile.zip, save it to the local machine at the location /opt/myfile.zip and unzip the file after download, the command is:

```
<Download>
   <URL>http://www.mycompany.com/myfile.zip</URL>
   <Destination>/opt/myfile.zip</Destination>
   <Action>UNZIP</ Action >
</Download>
```

When completed, the contents of the zip archive are located at "/opt/myfile".

One use of the <Download> command is to work around the 16kB startup package size limitation. For example, if you need to add several .jar files into the Wowza Server "lib" folder and these files push your startup package size over the 16kB limit, you might package these files into a separate zip archive. You can then host this zip archive on a web server and use the <Download> command to install the files into the Wowza Server "lib" folder.

For this example let's say we have two .jar files; wms-plugin-modulea.jar and wms-plugin-moduleb.jar. First, create the directory structure:

```
[wowzamodules]
   [lib]
      wms-plugin-modulea.jar
      wms-plugin-moduleb.jar
```

Next, zip up the [wowzamodules] folder into a zip archive named "wowzamodules.zip" and copy it to your company's web server. Let's assume this file is now available at http://www.mycompany/modules/wowzamodules.zip. The <Download> command to install this package into the Wowza Server lib folder is:

```
<Download>
   <URL>http://www.mycompany/modules/wowzamodules.zip</URL>
   <Destination>/opt/ wowzamodules.zip</Destination>
   <Action>INSTALL</ Action >
</Download>
```

Element <Download>/<URL>

The <URL> is the URL of the file to be downloaded. The download can be performed over SSL by starting the url with https:// rather than http://. The url can also contain query parameters. The file will be downloaded using the GET method unless <Data> is specified.

Element <Download>/<Data>

The Data is text data that will be included as part of the body of the HTTP request. You can use post data to send user name and password information to your web server so you can protect your content.

Element <Download>/<Header>: <Name> and <Value>

The <Header> elements are name value pairs added to the header part of the HTTP request. An example would be:

```
<Header>
  <Name>Content-type</Name>
  <Value>text/plain</Value>
</Header>
```

Element <Download>/<Destination>

The <Destination>element is the path to which the file will be saved (including the filename). This path can be relative or absolute. The base directory when calculating a relative file path, is the root directory of the startup package (the folder that contains the startup.xml file).

Element <Download>/<Action>

The <Action> element is the action performed after the file is download. The action can either be UNZIP or INSTALL. If the action is UNZIP the downloaded file will be unzipped using the unzip command. If the action is INSTALL the downloaded file will be unzipped and the contents of the folder will be installed (copied) into the Wowza Server installation folder.

Command <RunScript>

The <RunScript> command will execute a script on a running Amazon instance.

```
<RunScript>
  <Script>[relative-or-absolute-file-path]</Script>
  <Param>[parameter]</Param>
  <Param>[parameter]</Param>
</RunScript>
```

Element <RunScript>/<Script>

The <Script> element is the path to the script file to be executed. This path can be relative or absolute. The base directory when calculating a relative file path, is the root directory of the startup package (the folder that contains the startup.xml file).

Element <RunScript>/<Param>

The <Param> elements are parameters that will be passed to the running script. For example the following <RunScript> command:

```
<RunScript>
  <Script>scripts/copyfile.sh</Script>
  <Param>filea.txt</Param>
  <Param>fileb.txt</Param>
</RunScript>
```

Would be the equivalent of executing the command:

```
./scripts/copyfile.sh filea.txt fileb.txt
```

Before a script it executed, the startup processor initializes several environment variables with information that describes the current Amazon instance. These variables are:

```
AWSEC2_METADATA_INSTANCE_ID          - Amazon instance id
AWSEC2_METADATA_SECURITY_GROUPS      - Security group
AWSEC2_METADATA_LOCAL_IPV4           - Local IP address
AWSEC2_METADATA_AMI_LAUNCH_INDEX     - Launch index
AWSEC2_METADATA_PUBLIC_HOSTNAME      - Public host name
AWSEC2_METADATA_PRODUCT_CODES        - DevPay product code
AWSEC2_METADATA_INSTANCE_TYPE        - instance type (m1-small, m1-large, m1-xlarge)
AWSEC2_METADATA_HOSTNAME             - Public host name
AWSEC2_METADATA_LOCAL_HOSTNAME       - Local host name
AWSEC2_METADATA_PUBLIC_IPV4          - Public IP address
AWSEC2_METADATA_AMI_MANIFEST_PATH    - S3 manifest path
AWSEC2_METADATA_RESERVATION_ID       - Instance reservation ID
AWSEC2_METADATA_AMI_ID               - AMI ID
```

Additional Resources

Wowza Media Systems Amazon EC2 Support page:

http://www.wowzamedia.com/ec2support.html

Amazon Web Services: Resource Center:

http://developer.amazonwebservices.com/connect/kbcategory.jspa?categoryID=59

Amazon Web Services: Discussion Forums:

http://developer.amazonwebservices.com/connect/forumindex.jspa

Amazon EC2 Getting Started Guide:

http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/

Starting Amazon EC2 with Mac OS X (from Robert Sosinski):

http://www.robertsosinski.com/2008/01/26/starting-amazon-ec2-with-mac-os-x/

Firefox Extension for Amazon EC2:

http://www.wowzamedia.com/forums/showthread.php?t=794

Amazon S3 Firefox Organizer(S3Fox):

https://addons.mozilla.org/en-US/firefox/addon/3247

Cacti – Server monitoring/graphing tool

http://wwww.cacti.net/