



---

Wowza Media Server<sup>®</sup> Pro

# Wowza Pro Unlimited Amazon EC2 Edition

# **Wowza Media Server Pro: Wowza Pro Unlimited Amazon EC2 Edition**



**Version 1.1**

---

© Wowza Media Systems, Inc.  
<http://www.wowzamedia.com>

---

Copyright © 2006 - 2009 Wowza Media Systems, Inc. All rights reserved.

### **Third-Party Information**

This document contains links to third-party websites that are not under the control of Wowza Media Systems, and Wowza Media Systems 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 Media Systems provides these links only as a convenience, and the inclusion of any link does not imply that Wowza Media Systems 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 (c) 1999, 2006 Tanuki Software, Inc.

# Table of Contents

<b>Introduction .....</b>	<b>5</b>
<b>Starting a Wowza Pro Instance .....</b>	<b>5</b>
Starting an Instance .....	6
Opening an Telnet Session to an Instance .....	8
Stopping an Instance .....	8
<b>Wowza Pro Configuration .....</b>	<b>9</b>
Startup Package Example .....	9
Using a Startup Package .....	10
Startup Package Basics .....	10
Pre-built Startup Packages .....	11
<b>Wowza Pro Instance Details .....</b>	<b>11</b>
Wowza Pro Details .....	11
FTP Access .....	12
Java Management Extension (JMX) .....	13
Default Startup Package .....	14
Custom Module Development .....	14
Streaming Media Directly from S3 (beta) .....	14
<b>Startup Package Reference .....</b>	<b>16</b>
Command <Install> .....	16
Command <Download> .....	17
Command <RunScript> .....	19
<b>Additional Resources .....</b>	<b>20</b>

## Introduction

Amazon Elastic Cloud Computing (Amazon EC2) is a web service from Amazon that provides flexible, easy to provision computing resources. Wowza Media Server Pro (Wowza Pro) is a Java based interactive RTMP server for streaming audio, video and data to and from the Adobe Flash player client. This document describes how to install and configure Wowza Pro on Amazon EC2.

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

<http://www.amazon.com/ec2>

This document also assumes you are familiar with Wowza Pro. You can download a free Wowza Pro 10 edition license by going to:

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

The Wowza Pro download includes the Wowza Pro 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.

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

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

## Starting a Wowza Pro Instance

After signing up for Wowza Pro Unlimited for Amazon EC2 you will need to get the current Wowza Pro DevPay AMI IDs. You can get these IDs from the Wowza Pro EC2 Support page: <http://www.wowzamedia.com/ec2support.html>. These AMI IDs are what will be used to startup a Wowza Media Server Pro 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 2 regions "us-east-1" (US) and "eu-west-1" (Europe).

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

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.

Here is a similar guide for Mac OSX (by Robert Sosinski):

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

## **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 Pro 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:

## WOWZA PRO UNLIMITED AMAZON EC2 EDITION

```
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-----
MIIEBwIBAAKCAQEAKf63aqFW3oS3fh4MkTE3FSJecm/p+pLHoXBmOLYh1HF8LDwu8PgO4z6Qqx/
jCwZqiqW6lZJgA93KLr0yHc1+4maaE83Nitt+v1lly8+XGFbh2khVTBrGDqzXLKIkeRmx/imcXma
jdLJ6vo1NOWHBeyF+TWBXLsZyTmXGAPBHHIKfuN5bPXDFh5B/nFsbhLHmo5oINptVmy/eVYy65MY
olDir8kjan1Z2vSWbFuEYJAmWpFRBrlMfzt0I34+sfs01o1Wl+0cN+6b31trXHusUrEKqPbpU4r
XW+VuSt8yC2x+ByAr7fvuqEKNP8HESceJzNmw+yHpTzBisAIDNwkSwIDAQABAoIBAECNyQbEbxrt
PdkKTLVqcsUo/5+eaQC4Lsc0IfzeqrL0qXlZJDa+JivVerDqDMmTzQncoIZ8aF1DLaNNgkVWF5dk
aRnLM/vRNz2ohgTrYVZkq79Dk016uySxrSO4YG6WW8+HXLptjKLprAwIqkyMSSLA8pNPTx8x6i5N
MRTxddL5kFit2bnXsd1xmWDw4/FQtHvqz6gkhjylgkLdmX8+K/spw5O2ceInoDax3ml6/kpQmgUG
Zie/q9c9BSicVkcS4o13e5TRKzJ6TV5cjXRT59nexVYMxdcm7FuHI0b+BWMS4R3UIp5+t2nkUJFG
5H2TLUsXcMgC8PFWFY9Rz4o7KIECgYEA2Vym3R5nn73sp9h6gpQ4LMLXH6ci4oTIp4p7m+VVQT
YuI7nSt/32CTpzP0OGlj2nSybAR7U0RrJv0xcQaQNGbbfaRRdhlkbz+Iwim8TApwNLDBSRKa3rTP
lXGbf8IZCGpUlPEkgKscbvUnTotdk1hfBsZlTHFWcoZkhmcb5VkcGyEAq/Jqb68tidgytAEGu9wg
ZXdj7HgF57Iv4s1yusxywSOP4kJ16dVS0ynokP/TCm3jQW9WjdQHfuvEEQFNxpE3Kni+HYL4u8a2
C75rXHdvSxbH0WE2RP07EgoluFEgg4rOuqL1E2fQ9fiI4T/7vEkKRQp0Eu0XL/hSwKZJ+peTKMC
gYEAoGOMwWcZlbd8dFH5/Haw1ZxzgPnw08st2ZeZsXIO71nNwE6dnJRbtpHGAKAE+sdRpjySHiak
dcnro2Z7YCJusCwhlDE25wVPpgNkv5kaV6XwLAETbuUqoCEdpytgiCJIxHv88mZKACPDJGEDsyKF
Vx+DZzVoNCMBnOd0z/QvkmkCgYBxdHvsg48ttD0IbuDtlfq+Wl/TKQLkdbuJEHzSatKRS1V/utCo
4PxMnwt4mYS8dfP0miLnWe80PQbbHZwfGHRpDtCLEn9jD2PRbezTT29dAYQJUPaJHv9ZbpLhs3QL
yKoxTWlcrXeAQXNScpCZLf3pKQfKDBnPVUCnyOgb4VYmlwKBgHLk/iL9g/hkL+zz8o0p+SkzWYDH
CY1Gg97jUVpbEEAmbrEhM9wZPno9cCr1VWRHf5S9idsmbW+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.ppk”. 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.

Next, we are going to start a small Wowza Pro EC2 instance. Obtain the Wowza Pro 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 --region us-east-1 wowza-keypair -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 use 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 Pro 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 Pro 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.

## Opening an Telnet Session to an Instance

Before connecting to your Wowza Pro 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

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
```

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

Next, double click:

```
[install-dir]/examples/FastPlayVideoStreaming/client/fastplay.html
```

Enter the following information:

```
Server: rtmp://[instance-public-domain]/fastplay  
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.

## Wowza Pro Configuration

The method described above starts a generic Wowza Pro instance with all the example applications installed. You can configure a Wowza Pro 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.

### 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

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 Pro 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 Pro AMI. When the Wowza Pro 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 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 Pro 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 Pro 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/package-list.html>

Each package includes a REAMDE file that describes how to use the package.

## Wowza Pro Instance Details

A Wowza Pro 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 Pro Unlimited.

## Wowza Pro Details

Wowza Pro is installed at its default location:

```
/usr/local/WowzaMediaServerPro
```

The Wowza Pro service is running on the following ports:

```
1935 - rtmp port
80 - rtmp port
443 - rtmps port
8084 - JMX/JConsole Management
8085 - JMX/JConsole Management
21 - FTP access
22 - SSH access
```

These ports will need to be opened for the Wowza Pro 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
ec2-authorize default --region us-east-1 -p 8084
ec2-authorize default --region us-east-1 -p 8085
ec2-authorize default --region us-east-1 -p 21
ec2-authorize default --region us-east-1 -p 22
```

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 Pro 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 Pro 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 6970-9999 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 Pro 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 communication method (rather than the PASV). Consult your ftp client's documentation for more information.

The following Wowza Pro 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 Pro using ftp. When you log into a Wowza Pro 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
```

## Java Management Extension (JMX)

The JMX/JConsole interface (as described in the "Server Management Console and Monitoring" chapter of the Wowza Media Server Pro 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 Pro User's Guide:

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

## Default Startup Package

If you start the Wowza Pro 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 all of the examples that are included with Wowza Media Server Pro.

If you supply your own startup package by default there 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 (beta)

You can use Wowza Pro EC2 to stream media directly from the Amazon Simple Storage Service (S3). Wowza Pro 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 Pro DevPay AMI ID v EC2 1.1.10” or greater.

To use this feature, startup Wowza Pro EC2 using the vods3.zip startup package. You can download this package from the following url:

<http://wowzamediasystems.s3.amazonaws.com/com/wowza/startup/vods3.zip>

To startup an instance using this startup package execute the following command:

```
ec2-run-instances ami-123456 --region us-east-1 -t m1.xlarge -f vods3.zip
```

Where “ami-123456” is the current Wowza Pro 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.zip startup package is configured with S3 authorization turned off. This means that all content must be publicly available. To stream non-publicly available content, unzip the vod3s.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 vods3 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 vods3.zip startup package. To modify the default configuration edit the file “wowza/conf/StreamNameAliasMap.txt” that is included in the vods3.zip startup package and create a new startup package with the modifications. To do this, unzip the vods3.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/StreamNameAliasMap.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
```

## 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 Pro 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 Pro 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 Pro 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>
  <PostData>[data]</PostData>
  <Header><Key>[key-name]</Key><Value>[value]</Value></Header>
  <Header><Key>[key-name]</Key><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 Pro “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 Pro “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:

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

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

```
<Download>
  <URL>http://www.mycompany/modules/wowzmodules.zip</URL>
  <Destination>/opt/ wowzmodules.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 <PostData> is specified.

### **Element <Download>/<PostData>**

The PostData 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>: <Key> and <Value>**

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

```
<Header>
  <Key>Content-type</Key>
  <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 Pro 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>