Wowza Media Server® 3

Wowza Media Server for Amazon EC2 Edition

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Wowza Media Server for Amazon EC2 Edition

Version: 3.6.4

http://www.wowza.com

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

This document is for informational purposes only and in no way shall be interpreted or construed to create any warranties of any kind, either express or implied, regarding the information contained herein.

Third Party Information

This document contains links to third party websites that are not under the control of Wowza Media Systems, LLC ("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 provides 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.

This document refers to third party software that is not licensed, sold, distributed or otherwise endorsed by Wowza. Please ensure that any and all use of Wowza® software and third party software is properly licensed.

Trademarks

Wowza, Wowza Media Systems, Wowza Media Server and related logos are either registered trademarks or trademarks of Wowza Media Systems, LLC in the United States and/or other countries.

Amazon Web Services, AWS, Amazon EC2, EC2, Amazon Elastic Compute Cloud, Amazon S3, Amazon Simple Storage Service, Amazon DevPay, and DevPay are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

Adobe and Flash are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Microsoft and Silverlight are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

QuickTime, iPhone, iPad, and iPod touch are either registered trademarks or trademarks of Apple, Inc. in the United States and/or other countries.

Other product names, logos, designs, titles, words or phrases mentioned may be third party registered trademarks or trademarks in the United States and/or other countries.

Third party trademarks are used solely to identify and describe third party products as being compatible with Wowza products. Wowza is in no way sponsored, endorsed by or otherwise affiliated with any such third party trademark owners.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Third Party Copyright Notices

Apache Commons Lang: Copyright © 2001-2011, The Apache Software Foundation

Apache Commons Modeler Component: Copyright © 2002-2008, The Apache Software Foundation

Bouncy Castle Crypto APIs: Copyright © 2000 – 2008, Legion of the Bouncy Castle

Jackson JSON Parser: Copyright © 2009, FasterXML, LLC

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

Joda-Time version 2.1: Copyright © 2012, Joda.org.

Libgcc_s-4 library and Libstdc++ library: Copyright © 2011, Free Software Foundation, Inc.

LibVA libraries: Copyright © 2007, Intel Corporation. All rights reserved.

Log4j and Mina: Copyright © 2006, The Apache Software Foundation

NVIDIA Video Codec SDK: Copyright © 2013, NVIDIA Corporation

Silver Egg Technology: Copyright © 2001, Silver Egg Technology

Speex Codec: Copyright © 2002-2003, Jean-Marc Valin/Xiph.Org Foundation

VideoEncoderH264VAAPIlmpl: Copyright © 2012, Intel Corporation. All Rights Reserved.

Vorbis/Ogg libraries: Copyright © 2011, Xiph.Org Foundation

WebM VP8 Codec libraries: Copyright © 2010, Google Inc. All rights reserved.

Document History

Version	Description	Release date
Initial release	Initial document release	03-18-2011
3.0.0	Updated for Wowza Media Server 3.0.0	11-01-2011
3.0.4	Updated for Wowza Media Server 3.0.4	01-26-2012
3.1.2	Updated for Wowza Media Server 3.1.2	06-15-2012
3.5.0	Updated for Wowza Media Server 3.5.0	01-16-2013
3.6.1	Updated for Wowza Media Server 3.6.1	05-30-2013
3.6.2	Updated for Wowza Media Server 3.6.2	06-10-2013
3.6.3	Updated for Wowza Media Server 3.6.3	11-01-2013
3.6.4	Updated for Wowza Media Server 3.6.4	01-03-2014

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Table of Contents

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Introduction

This document describes how to install and configure Wowza Media Server 3.6 on Amazon Elastic Compute Cloud (Amazon EC2), an Amazon web service that provides resizable compute capacity in the cloud. Amazon EC2 is a cloud-computing platform that virtualizes computing resources as virtual machines. A single virtual machine configuration is registered as an *Amazon Machine Image* (AMI). Wowza Media Systems provides different Amazon Linux AMIs with preconfigured and tuned versions of Wowza Media Server that are ready to start using the Amazon Web Services (AWS) Management Console. You can use these AMIs as needed to launch Wowza Media Server for EC2 instances, paying for your running instance time and bandwidth consumption only while the instances are running. You can launch Wowza Media Server instances in specific geographical locations that are closer to your audience to reduce latency and in multiple geographical locations to provide high levels of redundancy.

Note

This document assumes that you're familiar with Wowza Media Server. If you aren't familiar with Wowza Media Server, you can get a free 30-day trial license for Wowza Media Server by completing a request form on the Wowza Media Server Trial webpage. The Wowza Media Server Trial Edition download includes the Wowza Media Server software, all premium AddOns, documentation, and examples. The Wowza Media Server User's Guide contains comprehensive documentation about how to install and use the server.

After you have your client-side and server-side applications up-and-running on your computer, use this document to learn how to deploy Wowza Media Server for Amazon EC2.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Getting Started

To get started using Wowza Media Server 3.6 for Amazon EC2, you must sign-in to or create an Amazon Web Services (AWS) account and then select one of the following licensing model options for each Wowza Media Server for EC2 instance that you plan to use.

Option 1: Bring Your Own (LicKey)

The LicKey License option enables you to use your regular Subscription (Daily or Monthly) or Perpetual license key with your Wowza Media Server for EC2 instance. When using this option, billing for your running instance time and bandwidth consumption is managed by Amazon. Wowza Media Server Subscription users will continue to receive a separate monthly invoice from Wowza for usage of the Wowza Media Server software and AddOns. This option provides access to all Wowza Media Server functionality and all premium AddOns, including Wowza Transcoder (for 64-bit instances only), Wowza nDVR, and Wowza DRM.

Option 2: DevPay

The DevPay License option enables you to use a Wowza Media Server software license that's embedded in a prebuilt Amazon DevPay AMI (a separate license isn't needed). This option provides the convenience of a combined monthly invoice from Amazon for running instance time, bandwidth consumption, and Wowza Media Server usage; however, it doesn't provide access to the premium AddOns.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Deploying Wowza Media Server for EC2 Instances

This section describes how to deploy Wowza Media Server 3.6 for EC2 instances using the Amazon Web Services (AWS) Management Console. EC2 Management Console is a web interface that enables you to manage Amazon EC2 and Wowza Media Server AMIs from a web browser. Many users find that it's easier to use the Management Console instead of the EC2 command line tools.

> **Important**
>
> This document describes Management Console functionality that's only supported by the Mozilla Firefox and Google Chrome web browsers.

To start the EC2 Management Console, go to https://console.aws.amazon.com/ec2 and sign in using the email address and password that you specified when you signed up for AWS.

The **EC2 Dashboard** will load in the EC2 Management Console. If a different dashboard is displayed, click the **Services** tab in the navigation bar at the top of the webpage, click **All AWS Services**, and then click **EC2**.

When you sign-in to the EC2 Management Console for the first time, the upper-right side of the navigation bar displays a drop-down list called **Select a Region**. Wowza provides public Amazon Linux AMIs that are preconfigured for specific regions. For more information about how to select a region for your Wowza Media Server for EC2 instance, see Choosing a Wowza Media Server AMI.

Choosing a Wowza Media Server AMI

Wowza provides public Amazon Linux AMIs with preconfigured versions of Wowza Media Server 3.6 that are ready to start through the EC2 Management Console. You must get a current Wowza Media Server AMI ID for each Wowza Media Server for EC2 instance that you plan to use.

Two collections of prebuilt AMIs are listed on the Wowza Prebuilt AMIs webpage, organized by licensing option. Before you choose an AMI from one of these collections for your instance, be sure to consider the following factors:

- The licensing model option that you want to use (either **LicKey** or **DevPay**). For more information about the licensing options, see the Wowza on Amazon AWS webpage.
- The region that you want to use. For help in selecting a region that meets your requirements, see Regions and Availability Zones.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Clicking a link for a Wowza Media Server AMI ID in the Wowza Prebuilt AMIs webpage launches the selected Wowza Media Server AMI in the EC2 Management console. For more information, see Launching the instance.

Creating a key pair

After you have decided on a Wowza Media Server 3.6 AMI to use, you should create a key pair for the region that you want to stream from. A key pair enables you to use management tools such as Telnet to connect to your Amazon EC2 instance after you launch it. Public AMI instances use a public/private key pair to log in instead of a password. The public key half of this key pair is embedded in your instance when you launch it, allowing you to use the private key to log in securely without a password.

Key pairs are regional resources and must be configured in each of the regions that you plan to use for streaming. You can create key pairs for a region at any time so that they'll be available to use when you launch your instance in that region. You can also create a key pair for a specific instance that you launch in the EC2 Management Console.

To create a key pair for use in a specific region at a later time, do the following:

1. Open the EC2 Management Console.
2. In the navigation bar, select the **Region** that you want to stream from.
3. In the **Navigation** pane, under **Network & Security**, click **Key Pairs**.
4. Click the **Create Key Pair** button.
5. In the **Create Key Pair** dialog box, in **Key pair name**, enter a name for your key pair (**[key-pair-name]**) and then click the **Yes** button.

You'll be prompted to open or save the private key (**[key-pair-name].pem**) file. Save the file to your computer as it's the only copy of your private key. Amazon doesn't store this key.

Opening ports for streaming

After you have decided on a Wowza Media Server 3.6 AMI to use, you should define firewall rules for your instances by configuring a security group for the region that you want to stream from. You can configure the **default** security group for the region that you want to stream from or you can create a new security group in that region. You must open several TCP and UDP ports in the firewall for streaming.

Security groups are regional resources and must be configured in each region that you plan to use for streaming. You can create security groups for a region at any time so that they'll be available to use when you launch your instance in that region. You can also create a security group for a specific instance that you launch in the EC2 Management Console.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

To configure a security group for use in a specific region at a later time, do the following:

1. Open the EC2 Management Console.
2. In the navigation bar, select the **Region** that you want to stream from.
3. In the **Navigation** pane, under **Network & Security**, click **Security Groups**.
4. (Optional) To create a new security group, click the **Create Security Group** button, provide a **Name** and **Description** in the dialog box, and then click the **Yes, Create** button.
5. In the list of security groups, select the security group that you want to configure for streaming.
6. In the lower pane, on the **Inbound** tab, create rules to open ports for streaming as well as ports for other connection protocols that enable you to manage the instance. For more information about the required and optional ports used by Wowza Media Server, see Ports Used for Streaming. To create a rule:
   a. Choose a protocol from the table below (for example, **RTMP**).
   b. In the **Create a new rule** list, select the rule type shown in the table for the protocol (for example, for the **RTMP** protocol select **Custom TCP rule**).
   c. In the **Port range** box, enter the port or port range value shown in the table for the protocol (for example, for the **RTMP** protocol enter **1935**).
   d. In the **Source** box, enter **0.0.0.0/0** (this is the default value).
   e. Click the **Add Rule** button.

Protocol	Rule to select	Port range to enter
RTMP	Custom TCP rule	1935
HTTP	Custom TCP rule	80
HTTPS, RTMPS	Custom TCP rule	443
RTSP	Custom TCP rule	554
RTP, MPEG-TS	Custom UDP rule	0-65535
FTP	Custom TCP rule	21
SSH and Telnet	Custom TCP rule	22
Wowza Stream Manager	Custom TCP rule	8086
JMX Interface	Custom TCP rule	8084-8085
Ping and Traceroute	All ICMP	Not applicable

7. Repeat step 6 to create additional rules. When you've finished creating rules for all of the protocols that you plan to use, click the **Apply Rule Changes** button.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Launching the instance

To launch your Wowza Media Server 3.6 for Amazon EC2 instance, do the following:

1. In the Wowza Prebuilt AMIs webpage, click the link for the Wowza Media Server AMI that you want to use.

2. If you're signed-in to the EC2 Management Console, the selected instance will launch in the Console and the **Choose Instance Type** page will be displayed.

   If you're not signed-in, provide the email address and password that you specified when you signed up for AWS, and then sign in.

   The **Choose Instance Type** page enables you to select the instance type that you want to use. The instance type specifies the hardware configuration for your EC2 instance. All Wowza Media Server AMIs are set to use the **m1.small** instance type by default. To select a different instance type, click **All instance types** in the navigation pane, and then click the instance type that you want to use in the list.

   If you're not sure which instance type to select, see Amazon EC2 Instance Types. Because pricing per instance-hour can vary based on the selected instance type, also review the Amazon EC2 Pricing page.

3. Click the **Next: Configure Instance Details** button to configure details for your selected instance type. The **Configure Instance** page enables you to configure optional settings for your selected instance type as well as to configure advanced instance options for loading a Wowza Media Server startup package to launch your Wowza Media Server instance with a customized configuration.

   To help you best configure the optional settings for your selected instance, click the **Information** (ⓘ) icon next to each option to learn more about it. Note that additional charges may apply for some of these options.

4. A Wowza Media Server startup package is a compressed (zipped) folder that contains a startup manifest file (**startup.xml**), configuration files, and scripts. It allows you to configure a Wowza Media Server instance at launch time by passing in user data. If you don't specify a startup package on this page, the default startup package is used. The **default** startup package contains the following streaming applications:

   - live
   - videochat
   - vod
   - vods3

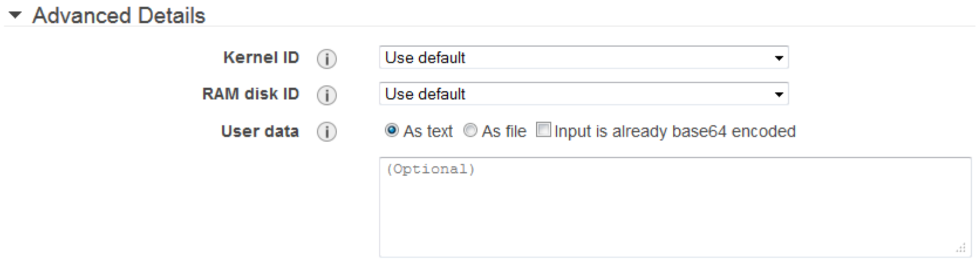
Note: Wowza Provides Prebuilt Startup Packages

You can download prebuilt startup packages from the Wowza Media Server for Amazon EC2: Startup Packages webpage and either use them "as-is" or modify them to meet your requirements. To learn more about startup packages and how to modify them to suit your needs, see Wowza Media Server Configuration Startup Packages.

If you load a prebuilt startup package that you've modified when launching an instance, the **default** startup package won't be used and your modified startup package must provide all of the application configurations that are required for streaming.

To load a Wowza Media Server startup package to launch your Wowza Media Server instance with a customized configuration, do the following:

a. On the **Configure Instance** page, click **Advanced Details** to reveal the advanced options:



Advanced Details

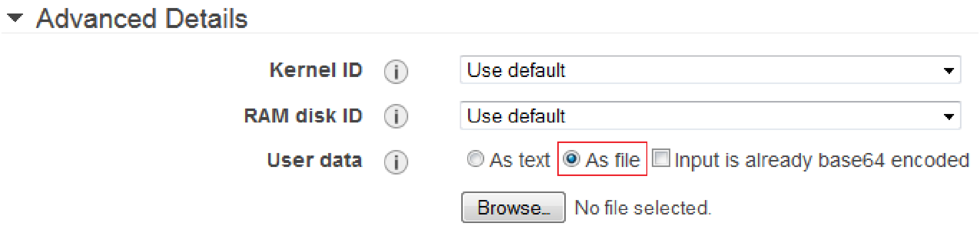
Kernel ID: Use default

RAM disk ID: Use default

User data: ● As text ○ As file □ Input is already base64 encoded

(Optional)

b. Under **User data**, select the **As file** option:



Advanced Details

Kernel ID: Use default

RAM disk ID: Use default

User data: ○ As text ● As file □ Input is already base64 encoded

Browse... No file selected.

c. Click the **Browse** button, and then find and upload the startup package (.zip) file in the **File Upload** dialog box.

Note:

Make sure that the **Input is already base64 encoded** checkbox is cleared.

5. Click the **Next: Add Storage** button to configure optional storage device settings that your instance is launched with. You can attach additional Elastic Block Store (EBS) volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. For more information about how to configure the **Add Storage** page, see Storage.

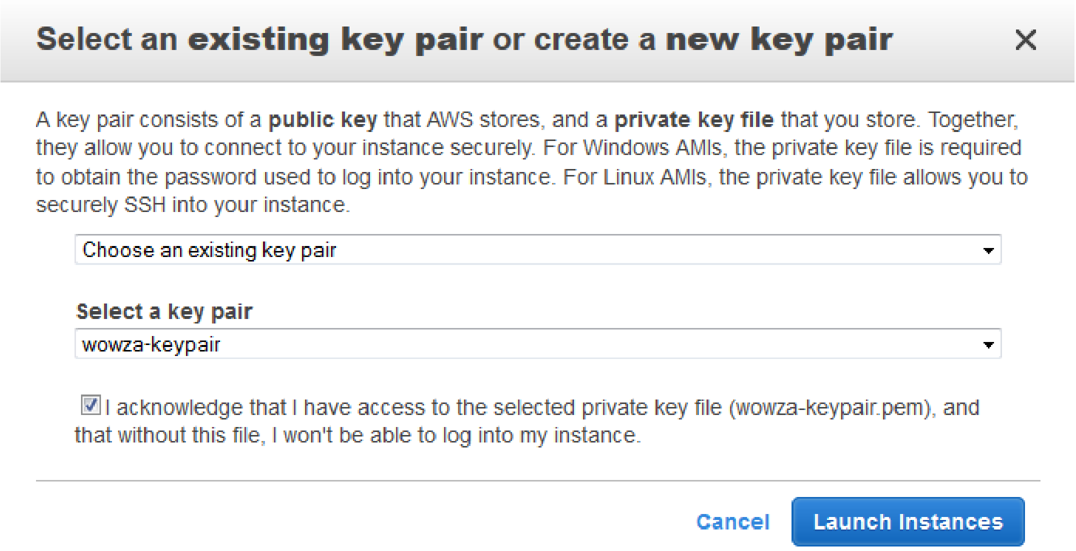
6. Click the **Next: Tag Instance** button to configure optional tagging options for your Amazon EC2 resources. For more information about how to configure the **Tag Instance** page, see Tagging Your Amazon EC2 Resources.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

7. Click the **Next: Configure Security Group** button to configure firewall rules that control the traffic for your instance. On the **Configure Security Group** page, you can either create a new security group for this instance or select a security group that you previously created. The security group must be configured for streaming. For more information, see Opening ports for streaming.

8. Click the **Review and Launch** button. On the **Review** page, review your instance details and then click the **Launch** button. This will display a dialog box where you can either select an existing key pair or create a new key pair for use with your instance.

   If you previously created a key pair for use with this instance, select **Choose an existing key pair** in the top list, select the key pair name in the **Select a key pair** list, select the **Acknowledgement** checkbox, and then click the **Launch Instances** button:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

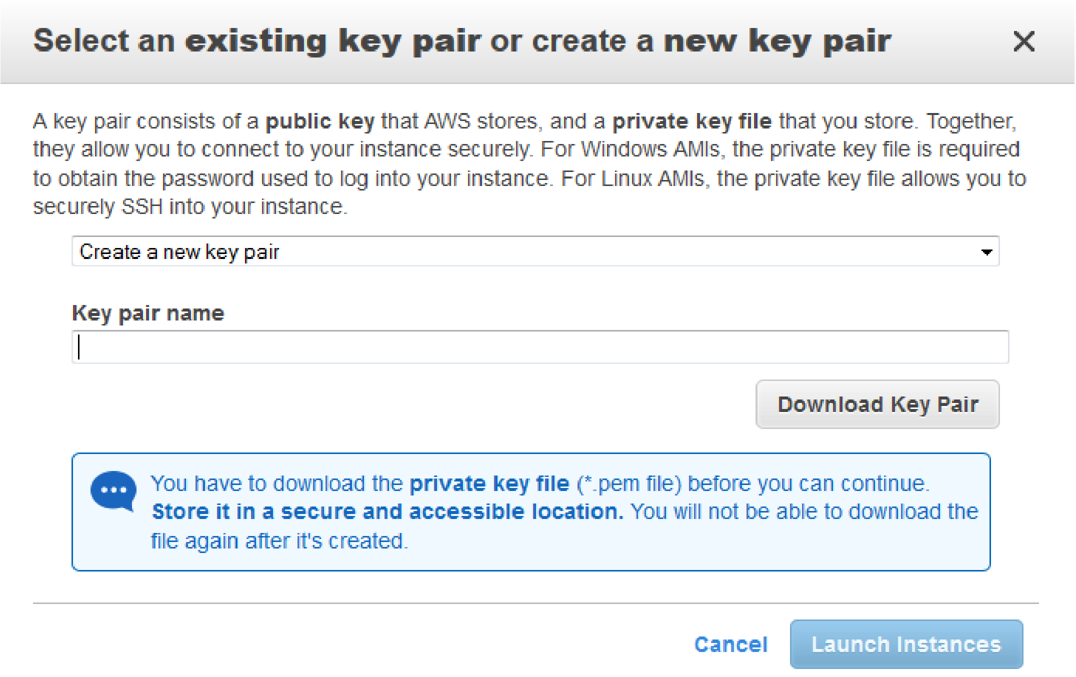
Choose an existing key pair

Select a key pair
wowza-keypair

☑ I acknowledge that I have access to the selected private key file (wowza-keypair.pem), and that without this file, I won't be able to log into my instance.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

To create a new key pair for use with this instance, do the following:



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Create a new key pair

Key pair name

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

a. In **Key pair name**, enter a name for your key pair (**[key-pair-name]**) and then click the **Download Key Pair** button.

b. You'll be prompted to open or save the private key (**[key-pair-name].pem**) file. Save the file to your computer as it's the only copy of your private key. Amazon doesn't store this key.

c. Click the **Launch Instances** button.

9. In the **Launch Status** page, click the **View Instances** button to see the status of your EC2 instance. It may take several minutes for your instance to launch. After the **Instance State** changes from **pending** to **running**, the instance is started. It might take an additional minute or two after that before Wowza Media Server is available for streaming.

Important

You'll start incurring charges for your running instance after it boots and charges will accrue for each hour or partial hour that you keep the instance running—even when it's idle. When you no longer need the instance, be sure to terminate it in order to stop incurring charges. For more information, see Terminating the instance.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Notes

If you use a **LicKey** AMI ID, the instance uses a temporary license key when you launch it for the first time. You must replace the temporary license key with a valid Wowza Media Server 3.6 Subscription (Daily or Monthly) or Perpetual license key. For more information, see Adding a Subscription or Perpetual license key.

If you use a **DevPay** AMI ID, an activation key isn't required to use the product. If you see a message that states that an activation key may be required, you should ignore it.

Getting the instance public domain name

You must use the public domain name (or hostname) of your running instance to access the instance remotely for streaming and to connect to the instance using Secure Shell (SSH). To get the public domain name of your instance, do the following:

1. In the **Navigation** pane of the EC2 Management Console, under **Instances**, click **Instances**.
2. Select the running instance.
3. In the lower pane, click the **Description** tab. The **Public DNS** value is the public domain name of your running instance.

Note

If you require a persistent public IP address that can be assigned to and removed from instances as necessary, you can use a static IP address that's designed for dynamic cloud computing called an *Elastic IP address*. An Elastic IP address is associated with your account (not with a particular instance). For more information, see Elastic IP Addresses (EIP).

Testing the instance

You can quickly verify that your running Wowza Media Server 3.6 for EC2 instance is working correctly by using an example player on a remote computer to stream a sample video on-demand file from the instance. To stream the sample on-demand video file from your running instance, do the following:

1. Download the Wowza Media Server 3.6 example files.
2. Extract the example files from the compressed (zipped) folder and then run the **VideoOnDemandStreaming** example installation batch file to install the example player:

```
[install-dir]/examples/VideoOnDemandStreaming/install.bat
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

3. Open the **FlashRTMPPlayer** example player in a web browser:

```
[install-dir]/examples/VideoOnDemandStreaming/FlashRTMPPlayer/player.html
```

4. In the **FlashRTMPPlayer** example player, enter the following information and then click the **Connect** button:

```
Server: rtmp://[instance-public-dns]/vod
Stream: mp4:sample.mp4
```

   Where **[instance-public-dns]** is the public domain name of your Wowza Media Server for EC2 instance.

The **Big Buck Bunny** video should start to play.

Managing the instance over an SSH session

You may need to open a secure Telnet session to your Amazon EC2 instance using Secure Shell (SSH) in order to start and stop the Wowza Media Server or to make configuration changes. Public AMI instances use a public/private key pair to log in instead of a password. The public key half of this pair is embedded in your instance, allowing you to use the private key half to log in securely without a password. You can use the key pair that you created for the region in which your Amazon EC2 instance is running (see Creating a key pair).

On Windows operating systems, you can open a secure Telnet session to your Amazon EC2 instance by using the PuTTY Secure Shell client for Telnet. This section describes how to use the PuTTY client and the PuTTYgen key generator, which you can download from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.

> **Note**
>
> For additional details about the procedures in this section, see Connecting to Linux/UNIX Instances from Windows Using PuTTY.

Converting your private key

The PuTTY Secure Shell client for Telnet doesn't natively support the private key format generated by Amazon EC2. Fortunately, PuTTY has a tool called PuTTYgen that you can use to convert your private key to the required **PuTTY Private Key File (*.ppk)** format. To convert the **[key-pair-name].pem** file that you created to a **[key-pair-name].ppk** file, do the following:

1. Start **PuTTYgen** (**Start** > **All Programs** > **PuTTY** > **PuTTYgen**).
2. In the **PuTTY Key Generator** dialog box, click **Load** and select the **[key-pair-name].pem** file that you want to convert. You'll need to select the **All Files *.*** option in the **File filter** drop-down list to see PEM files in the file list.
3. Click **Save private key** and save the file with the name **[key-pair-name].ppk**.
4. Close the dialog box.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Connecting to the instance

To open a Secure Shell (SSH) session to your Amazon EC2 instance, do the following:

1. Start **PuTTY** (**Start** > **All Programs** > **PuTTY** > **PuTTY**).
2. In the **PuTTY Configuration** dialog box, in the **Category** tree, select **Session**.
3. On the **Basic options for your PuTTY session** page, in **Specify the destination you want to connect to**:
    - In the **Host Name (or IP address)** field, enter **[instance-public-dns]**, where **[instance-public-dns]** is the public domain name of your Amazon EC2 instance running Wowza Media Server 3.6.
    - In the **Port** field, enter **22**. Make sure that you have followed the instructions for opening TCP port 22 in Opening ports for streaming.
    - Under **Connection type**, select the **SSH** option.
4. In the **Category** tree, select **Connection** > **SSH** > **Auth**.
5. On the **Options controlling SSH authentication** page, click the **Browse** button to find and open the **[key-pair-name].ppk** file.
6. (Optional). If you will be opening this same session later, you can save it for future use. To save the session information:
    a. Select **Session** in the **Category** tree.
    b. On the **Basic options for your PuTTY session** page, enter a name for the session in **Saved Sessions**, and then click the **Save** button.
7. On the **Options controlling SSH authentication** page, click the **Open** button to open the secure Telnet session. The first time you connect to your instance, you'll get a **PuTTY Security Alert** that references the first use of **[key-pair-name]**. Click **Yes** to accept the security key.

If you previously saved the SSH session information for this Amazon EC2 instance, do the following:

1. Start **PuTTY** (**Start** > **All Programs** > **PuTTY** > **PuTTY**).
2. In the **PuTTY Configuration** dialog box, in the **Category** tree, select **Session**.
3. On the **Basic options for your PuTTY session** page, in **Load, save or delete a stored session**, click the saved SSH session name and then click the **Load** button.
4. Click the **Open** button to open the secure Telnet session.

You should see the **login as:** prompt in the Telnet window. Enter the username **ec2-user** to login to your Amazon EC2 instance. When working with Wowza Media Server, it's best to be logged in as the **root** user. You can switch to the **root** user by entering the following command in the Telnet window:

```
sudo su –
```

Disconnecting from the instance

To end your SSH session, enter the **exit** command or press CTRL+D. You may have to do this twice if you're logged-in as the **root** user.

Important

Disconnecting from the instance doesn't affect the instance status. If it was running when you disconnected, it'll continue to run and you'll continue to incur charges for your running instance. When you no longer need the instance, be sure to terminate it in order to stop incurring charges. For more information, see Terminating the instance.

Uploading files to the instance via FTP

You may need to connect to your Wowza Media Server 3.6 instance using FTP to upload media files or Synchronized Multimedia Markup Language (SMIL) playlist files (for adaptive bitrate streaming), or to refresh configuration files. For convenience, most of the Wowza Media Server folders are symbolically linked to the **/home/wowza** folder for easy access using FTP.

Wowza Media Server instances come preinstalled with the FTP Server (**vsftpd**) for Linux. A default **wowza** FTP user account has been added to the system with the password set to the instance ID (this is done for security reasons). You can get the instance ID of a running instance either through the EC2 Management Console or, if logged-in to the instance, by using the following command in the Telnet window:

```
wget -q -O - http://169.254.169.254/latest/meta-data/instance-id
```

For security reasons, we strongly recommend that you change the default password for the **wowza** FTP user account for your AMI. To change the password, log-in to the instance as the **ec2-user** user, execute the following command, and then follow the prompts:

```
sudo passwd wowza
```

Note

Be sure to open TCP port 21 in your **Security Groups** settings so that you can connect to your instance using FTP. (See Opening ports for streaming.) You must also configure your FTP client to use **PORT** communication (ACTIVE mode). The FTP configuration doesn't support **PASV** communication (PASSIVE mode). Consult your FTP client documentation for more information.

Adding a Subscription or Perpetual license key

If you use a **LicKey** AMI ID, the instance uses a temporary license key when you launch it for the first time. You must replace the temporary license key with a valid Wowza Media Server 3.6 Subscription (Daily or Monthly) or Perpetual license key.

> **Note**
>
> If you're using a **DevPay** AMI ID, you don't need to change the license key.

You can use the following options to replace the temporary license key:

Telnet connection

1. Launch your Wowza Media Server for Amazon EC2 instance. (See Launching the instance.)
2. After the instance has started, connect to it using a secure Telnet session. (See Managing the instance over an SSH session.)
3. In the Telnet window, change directory to the **/usr/local/WowzaMediaServer/conf** folder. (See Linux shell command cheat sheet for beginners.)
4. Open the **Server.license** file in a text editor, replace the entire contents of the file with your Subscription (Daily or Monthly) or Perpetual license key value, and then save the file.
5. Restart the Wowza Media Server instance to activate the key. You can do this by executing the following commands:

   ```
   service WowzaMediaServer stop
   service WowzaMediaServer start
   ```

Startup package

You can include your license key in the **Server.license** file and include the file in a startup package. The license key value will then be passed-in as user data to replace the temporary license key when the instance is started. For more information about how to use startup packages, see Wowza Media Server Configuration Startup Packages.

FTP

You can connect to the instance via FTP and then upload a copy of the **Server.license** file that has the new license key to replace the existing file. For more information, see Uploading files to the instance via FTP.

Custom AMI

You can create a custom Amazon EC2 AMI that includes the license key. This option is for advanced users and instructions about how to do this is beyond the scope of this document.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Terminating the instance

When you terminate an instance, you'll lose all changes or files that you have on the server. If you have anything that you don't want to lose, be sure to save it to Amazon Simple Storage Service (Amazon S3) or to Amazon Elastic Block Store (EBS) before terminating the instance or you'll lose data.

After you've saved your data, do the following to terminate an instance:

1. In the **Navigation** pane, under **Instances**, click **Instances**.
2. Select the running instance(s) that you want to terminate.
3. Click the **Actions** button, and then click **Terminate**. The **Instance State** column for the selected instance(s) will show **shutting-down** and then **terminated**.

Important

Amazon recommends that you confirm that the machine reaches the **terminated** state before you sign out. You'll continue to be charged for instances that fail to shut down correctly.

Performance benchmarks

Below are some performance benchmarks when using Wowza Media Server 3.6 on Amazon EC2. These are total bitrate values that a single instance of a given instance type can handle for outgoing streams. To calculate concurrent connections, divide these numbers by the bitrate of your stream, in kilobits-per-second (Kbps).

```
m1.small:     150,000 Kbps
m1.large:     250,000 Kbps
m1.xlarge:    350,000 Kbps
```

For example if you're doing live streaming using a 500 Kbps live stream, then a given instance type can handle the following number of concurrent connections.

```
m1.small:     300 concurrent connections
m1.large:     500 concurrent connections
m1.xlarge:    700 concurrent connections
```

Linux shell command cheat sheet for beginners

Linux users connect to an instance using an SSH session. This section describes some useful commands for users who aren't familiar with the Linux environment. If the command you're looking for isn't included in this section, there are many Linux guides available on the Internet that you can consult.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Viewing directory contents

The following command changes your current directory to one that shows the most common user accessed Wowza Media Server directories:

```
cd /home/wowza
```

From here, you can display the subdirectories in the Wowza Media Server directory by typing the following command:

```
ls
```

To view the contents of a subdirectory, type the following:

```
cd [directory name]
```

To move up one level from the current directory, type the following:

```
cd..
```

Viewing log messages

You can interactively log entries as they are added to Wowza Media Server logs by executing the following commands:

```
cd /usr/local/WowzaMediaServer/logs
tail -f wowzamediaserver_access.log
```

Stopping Wowza Media Server

To stop Wowza Media Server, enter the following command:

```
service WowzaMediaServer stop
```

Starting Wowza Media Server

To start Wowza Media Server, enter the following command:

```
service WowzaMediaServer start
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Wowza Media Server Configuration Startup Packages

The previous section described how to launch a generic Wowza Media Server 3.6 for Amazon EC2 instance with most of the example applications installed. You can configure a Wowza Media Server instance at launch time by passing in user data in the form of a startup package. A startup package is a compressed (zipped) folder that contains a startup manifest file (**startup.xml**), configuration files, and scripts. Amazon EC2 limits the size of a startup package to 16 kilobytes (KB), which is the maximum user data size.

Wowza provides prebuilt startup packages that you can download from the Wowza Media Server for Amazon EC2: Startup Packages webpage. You can use them "as-is" or you can modify them to meet your requirements.

Startup package example

The following example shows the file structure of a simple startup package:

```
[mywowzaconfig]
        startup.xml
        [wowza]
                [applications]
                        [myapp]
                [conf]
                        Server.license
                        [myapp]
                                Application.xml
        [tuning]
                tune.sh
                MediaCache.xml
                setenv.sh
```

To see how a startup package is constructed, download and extract the contents of the Wowza Media Server **default** startup package: Download default_3.6.zip

Startup package basics

The startup manifest file (**startup.xml**) for the startup package outlined above has the following content:

```
<Startup>
        <Commands>
                <Install>
                        <Folder>wowza</Folder>
                </Install>
                <RunScript>
                        <Script>tuning/tune.sh</Script>
                </RunScript>
        </Commands>
</Startup>
```

In **startup.xml**, the **<Install>** command instructs the startup processor to copy the contents of the included **wowza** folder into the **/usr/local/WowzaMediaServer** folder of the running Wowza Media Server instance:

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

This gives you a chance to create application folders and configuration folders and files as well as inject a **Server.license** file for a **LicKey** instance.
The **<RunScript>** command instructs the startup processor to run the included script **tuning/tune.sh**:

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

This script changes configuration parameters based on instance size. This script must be a shell script (not a binary application) and will be executed by the operating environment that's running on the Wowza Media 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).

For more information about the commands that can appear in a startup manifest file (**startup.xml**), see the Startup Package Reference at the end of this document.

Prebuilt startup packages

Wowza provides several prebuilt startup packages that you can download from the Wowza Media Server for Amazon EC2: Startup Packages webpage. You can use them "as-is" or you can modify them to meet your requirements.

Default startup package

If you launch a Wowza Media Server for EC2 AMI without specifying a startup package, the **default** startup package is used. You can download and look at the **default** startup package:
Download default_3.6.zip
The following application names are configured in the **default** startup package:

- live
- videochat
- vod
- vods3

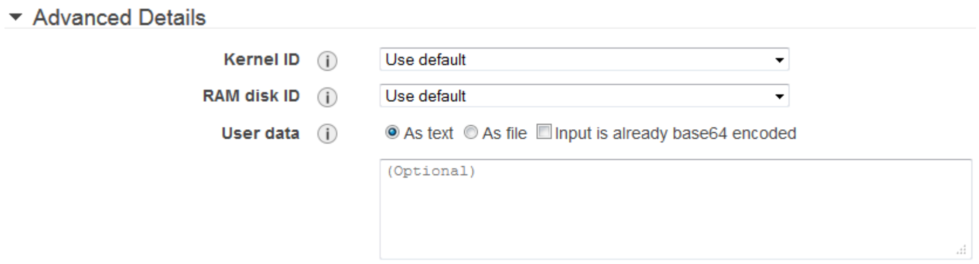
If you load your own startup package when launching an instance, the **default** startup package won't be used. Your startup package must provide all of the application configurations that are required for streaming.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Loading a startup package

When you launch a new instance in EC2 Management Console, you can load a Wowza Media Server startup package to launch the instance with a customized Wowza Media Server configuration. Wowza provides prebuilt startup packages that you can download from the Wowza Media Server for Amazon EC2: Startup Packages webpage. You can download a startup package and load it "as-is" or you can modify it to meet your requirements before you load it. To load a startup package, do the following:

a. On the **Configure Instance** page in the EC2 Management Console, click **Advanced Details** to reveal the advanced options:



Advanced Details

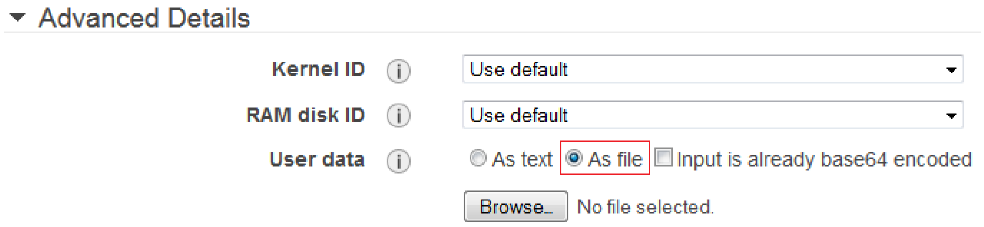
Kernel ID: Use default

RAM disk ID: Use default

User data: ● As text ○ As file □ Input is already base64 encoded

(Optional)

b. Under **User data**, select the **As file** option:



Advanced Details

Kernel ID: Use default

RAM disk ID: Use default

User data: ○ As text ● As file □ Input is already base64 encoded

Browse... No file selected.

c. Click the **Browse** button, and then find and upload the startup package (.zip) file in the **File Upload** dialog box.

Notes:

- Make sure that the **Input is already base64 encoded** checkbox is cleared.
- The ability to add startup package user data in EC2 Management Console is limited to Mozilla Firefox and Google Chrome web browsers.

Startup package debugging

The best way to debug a startup package is to launch an AMI that's using it and then look in the startup log file for errors or warnings. The startup log file is written to the following location:

```
/usr/local/WowzaMediaServer/logs/wowzamediaserver_startup.log
```

The log is quite extensive and should provide ample information to help debug startup package issues.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Streaming Media from Amazon S3

You can use Wowza Media Server 3.6 for Amazon EC2 instances to stream media directly from Amazon Simple Storage Service (Amazon S3). Wowza Media Server uses MediaCache caching technology to improve performance when streaming media from Amazon S3. The **MediaCache AddOn** package is a read-through caching mechanism for video on demand streaming that can pull content from an HTTP origin or from network-attached storage (NAS). The MediaCache system is tuned on a per-instance basis in Wowza Media Server startup packages. You can see the tuning in the **tuning/tune.sh** file that's included in any Wowza Media Server startup package.

This feature is included in the **default** startup package for Wowza Media Server for Amazon EC2 Edition and is available when using the **vods3** application.

To stream content from Amazon S3, use stream names in the following form. The **amazons3/** part of the stream name identifies that the stream is sourced from Amazon S3:

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

Using the above example, to play the **mycoolvideo.m4v** file that's stored in the Amazon S3 bucket **mybucket** at the path **videos/coolvideos**, the stream URLs are:

Adobe® Flash® Player (RTMP)

```
Server: rtmp://[instance-public-dns]/vods3
Stream: mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v
```

Adobe Flash Player (RTMP single URL)

```
rtmp://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v
```

Adobe Flash Player (Flash HDS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v/manife
st.f4m
```

Apple® iOS device (Apple HLS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v/playli
st.m3u8
```

Microsoft® Silverlight® (Smooth Streaming)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v/Manife
st
```

RTSP/RTP player or device

```
rtsp://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v
```

> **Note**
>
> When streaming using a stream name that includes path elements (for example, **mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4v**), you must supply both the **application** name and **applicationInstance** name as part of the URL. The example URLs above use the default **applicationInstance** name **_definst_**.

Amazon S3 authorization

By default the **vods3** application is configured with S3 authorization turned off. This means that all content must be publicly available. To stream non-publicly available content, extract the startup package from the **default_3.6.zip** compressed (zipped) folder and modify the system properties **awsAccessKeyId** and **awsSecretAccessKey** in **tuning/MediaCache.xml**. For example:

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

Set these two values to the **Access Key ID** and **Secret Access Key** for the user that you want to give access to, and then zip the default folder into a new compressed (zipped) folder. You can then use this new zipped folder as your startup package.

Amazon S3 stream name alias

By default, any publicly available content in Amazon S3 or in a public web server can be re-streamed through your instance. To limit the content to specific S3 buckets, you can use the stream name alias system to add aliases for the S3 buckets that you want to stream from. The **StreamNameAlias** AddOn package is included in the **default_3.6.zip** startup package to enable you to do this.

To modify the default configuration, extract the startup package from the **default_3.6.zip** compressed (zipped) folder, edit the **wowza/conf/aliasmap.play.txt** file that's included in the startup package, and then then zip the default folder into a new compressed (zipped) folder to create a new startup package that has the modified file.

When you edit the **wowza/conf/aliasmap.play.txt** file, you'll add wildcard entries for the S3 buckets that you want to stream content from and comment-out the default stream alias

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

rule. For example, to limit streaming to content coming from the S3 bucket **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 **mycoolvideo.m4v** file that's stored in the S3 bucket **mybucket** at the path **videos/coolvideos**, the stream URLs are:

Adobe Flash Player (RTMP)

```
Server: rtmp://[instance-public-dns]/vods3
Stream: mp4:mybucket/videos/coolvideos/mycoolvideo.m4v
```

Adobe Flash Player (RTMP single URL)

```
rtmp://[instance-public-
dns]/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v
```

Adobe Flash Player (Flash HDS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v/manifest.f4m
```

Apple iOS device (Apple HLS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v/Manifest
```

RTSP/RTP player or device

```
rtsp://[instance-public-
dns]/vods3/_definst_/mp4:mybucket/videos/coolvideos/mycoolvideo.m4v
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Sample stream name for vods3 application testing

Wowza provides a sample MP4 file at the following stream name that you can use to test your **vods3** application:

```
mp4:amazons3/wowzamediacache/sample/sample.mp4
```

Using the above stream name, to play the **sample.mp4** file that's stored in the Amazon S3 bucket **wowzamediacache** at the path **sample/**, the stream URLs are:

Adobe Flash Player (RTMP)

```
Server: rtmp://[instance-public-dns]/vods3
Stream: mp4:amazons3/wowzamediacache/sample/sample.mp4
```

Adobe Flash Player (RTMP single URL)

```
rtmp://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/wowzamediacache/sample/sample.mp4
```

Adobe Flash player (Flash HDS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/wowzamediacache/sample/sample.mp4/manifest.f4m
```

Apple iOS device (Apple HLS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/wowzamediacache/sample/sample.mp4/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/wowzamediacache/sample/sample.mp4/Manifest
```

RTSP/RTP player or device

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/wowzamediacache/sample/sample.mp4
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Wowza Media Server Instance

A Wowza Media Server 3.6 for Amazon EC2 instance is built using the Amazon Linux AMI as the base. This AMI is provided by Amazon.

Other components that are installed:

- Wowza Media Server 3.6
- The latest Java Development Kit (JDK) for Linux
- FTP Server (vsftpd) for Linux
- S3FS FUSE-based file system

Note

A web server isn't included in Wowza Media Server for Amazon EC2 AMIs.

Wowza Media Server details

Wowza Media Server 3.6 is installed at its default location:

```
/usr/local/WowzaMediaServer
```

The **Wowza Media Server 3.6.x** service runs on the following ports:

```
TCP 1935                - RTMP (all variants), RTSP, Smooth and Cupertino Streaming
TCP 80                  - RTMP (all variants), RTSP, Smooth and Cupertino Streaming
TCP 443                 - RTMP (all variants), RTSP, Smooth and Cupertino Streaming
TCP 554                 - RTMP (all variants), RTSP, Smooth and Cupertino Streaming
UDP 0-65535             - RTP and MPEG-TS UDP streaming
```

Wowza Media Server is managed using the following ports:

```
8084                    - JMX/JConsole Management
8085                    - JMX/JConsole Management
8086                    - Stream Manager and Administration
21                      - FTP access
22                      - SSH access
```

The Administration TCP port 8086 is configured to return load-balancing information over HTTP. What this means is that if you open a web browser and enter the following URL:

```
http://[instance-public-dns]:8086/connectioninfo
```

Where **[instance-public-dns]** is the public domain name for the instance.
Wowza Media Server will return load information for the instance. For example:

```
server=864
```

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

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Java Management Extensions (JMX)

The Java Management Extensions (JMX) interface to your instance is preconfigured to listen to connections on TCP ports 8084 and 8085 using the public domain name. You must open these ports to TCP traffic to connect to your EC2 instance successfully.
The JMX URL is:

```
service:jmx:rmi://[instance-public-dns]:8084/jndi/rmi://[instance-public-
dns]:8085/jmxrmi
```

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

```
[instance-public-dns]:8085
```

Where **[instance-public-dns]** is the public domain name of the instance. The default username is **wowza** and the default password is the instance ID. User access is managed in the following files:

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

Custom module development

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

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

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

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
com.wowza.amazonaws.ec2.AWSEC2_METADATA_ANCESTOR_AMI_IDS       - List of ancestor
                                                                 IDs
com.wowza.amazonaws.ec2.AWSEC2_METADATA_KERNEL_ID              - Kernel ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AVAILABILITY_ZONE      - Availability zone
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_KEYS            - Public keys
com.wowza.amazonaws.ec2.AWSEC2_METADATA_RAMDISK_ID             - RAM disk ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_BLOCK_DEVICE_MAPPING   - block mapping
```

Note

To learn more about Wowza Media Server module development, download and install the Wowza IDE and read the included User's Guide.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Startup Package Reference

This section describes the commands that can appear in a startup manifest file (**startup.xml**). The three commands are **<Install>**, **<Download>**, and **<RunScript>**.

Command <Install>

The **<Install>** command copies the contents of a folder that's contained in the startup package to the Wowza Media Server installation folder **/usr/local/WowzaMediaServer**.

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

Element <Install>/<Folder>

The **<Folder>** element specifies a folder in the startup package that's copied to the Wowza Media Server installation folder. The folder's directory structure should be the same as the Wowza Media Server installation folder. The specified directory path is relative to the startup package root.

For example, if you have a startup package with the following structure:

```
[startup-package]
       startup.xml
       [wowza]
              [applications]
                     [myapp]
              [conf]
                     Server.license
                     [myapp]
                            Application.xml
       [tuning]
              tune.sh
              MediaCache.xml
              setenv.sh
```

And the **startup.xml** file content is:

```
<Startup>
       <Commands>
              <Install>
                     <Folder>wowza</Folder>
              </Install>
              <RunScript>
                     <Script>tuning/tune.sh</Script>
              </RunScript>
       </Commands>
</Startup>
```

The contents of the **wowza** folder are copied to the **/usr/local/WowzaMediaServer** folder of the running Wowza Media Server instance. In this example, the result is that the application **myapp** is created and it will use the configuration file **wowza/conf/myapp/Application.xml**.

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Note

A startup packages must not exceed 16 kilobytes (KB) after it's compressed into a zipped folder (.zip). The **<Install>** command works well if you have a small number of configuration files. If you have a more extensive configuration that pushes the startup package size beyond the 16 KB limit, use the **<Download>** command described in the next section.

Command <Download>

The **<Download>** command downloads content from a web server and saves it to the local instance. The **<Download>** command includes the following elements:

```
<Download>
       <URL>[URL]</URL>
       <Data>[data]</Data>
       <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 required elements are **<URL>** and **<Destination>**. To download **http://www.mycompany.com/myfile.zip**, save it to the local computer at the location **/opt/myfile.zip**, and then unzip the downloaded file, the command is:

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

When completed, the extracted contents are located at **/opt/myfile**.

One use of the **<Download>** command is to work around the 16 kilobyte (KB) startup package size limitation. For example, if you have a more extensive configuration or you need to add several **.jar** files to the Wowza Media Server **lib** folder and these files push your startup package size over the 16 KB limit, you might package these files into a separate compressed (zipped) folder. You can then host the zipped folder on a web server and use the **<Download>** command to install the files to the Wowza Media Server **lib** folder.

For this example, let's say that you have two **.jar** files (**wms-plugin-modulea.jar** and **wms-plugin-moduleb.jar**) and that you want to set up two applications (**live** and **vod**). Let's also assume that you're using the **LicKey** licensing option, which requires you to supply your own **Server.license** file. First, create the following directory structure:

```
[wowzamodules]
       [applications]
              [live]
              [vod]
       [conf]
              Server.license
              [live]
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

```
                     Application.xml
              [vod]
                     Application.xml
       [lib]
              wms-plugin-modulea.jar
              wms-plugin-moduleb.jar
```

Next, place the **[wowzamodules]** folder into a compressed (zipped) folder named **wowzamodules.zip** and copy it to your company's web server. Let's assume that this file is now available at **http://www.mycompany.com/modules/wowzamodules.zip**. The **<Download>** command to install this package into the Wowza Media Server **lib** folder is:

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

Element <Download>/<URL>

The **<URL>** element is the URL of the file to be downloaded. The download can be performed over Secure Sockets Layer (SSL) by using the URL prefix **https://** instead of **http://**. The URL can also contain query parameters. The file is downloaded using the GET method, unless **<Data>** is specified.

Element <Download>/<Data>

The **<Data>** element is text data that's included as part of the HTTP request body. You can use POST data to send username and password information to your web server so that you can protect your content.

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

The **<Header>** elements are name-value pairs that are added to the header part of the HTTP request. For example:

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

You can use **<Header>** data to protect your content. For example, you can use the header values to specify a username and password using BASIC authentication:

```
<Header>
       <Name>Authorization</Name>
       <Value>Basic dXNlcm5hbWU6cGFzc3dvcmQ=</Value>
</Header>
```

Element <Download>/<Destination>

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

Element <Download>/<Action>

The **<Action>** element defines the action performed after the file is downloaded. The action can be either **UNZIP** or **INSTALL**. If the action is **UNZIP**, the downloaded file is unzipped using the UNZIP command. If the action is **INSTALL**, the downloaded file is unzipped and the folder contents are copied to the Wowza Media Server installation folder **/usr/local/WowzaMediaServer**.

Command <RunScript>

The **<RunScript>** command executes a script on a running instance. For example:

```
<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. When calculating a relative filepath, the base directory 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 are 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>
```

Is the equivalent of executing the following command:

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

Before a script is executed, the startup processor initializes the following environment variables with information that describes the current instance:

```
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
AWSEC2_METADATA_AVAILABILITY_ZONE      - Availability zone
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

```
AWSEC2_METADATA_PUBLIC_KEYS               - Public keys
AWSEC2_METADATA_ANCESTOR_AMI_IDS          - Ancestor AMI IDs
AWSEC2_METADATA_RAMDISK_ID                - RAM disk ID
AWSEC2_METADATA_BLOCK_DEVICE_MAPPING      - Block device mapping
AWSEC2_METADATA_KERNEL_ID                 - Kernel ID
```

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.

Additional Resources

- Amazon EC2 Overview: http://aws.amazon.com/ec2/
- Amazon Web Services Support Center: http://aws.amazon.com/support/
- Wowza for Amazon EC2 Support Center: http://www.wowza.com/forums/content.php?7

Copyright © 2006 - 2014 Wowza Media Systems, LLC. All rights reserved.