Wowza Streaming Engine™ for Amazon EC2

User's Guide

Wowza Streaming Engine for Amazon EC2 User's Guide

Version: 4.0

---

http://www.wowza.com

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.

No Endorsement or Warranty for Third-Party Links and Software

This document contains links to third-party websites ("Linked Sites") that are not under the control of Wowza® Media Systems, LLC ("Wowza"). Wowza is not responsible for the content on or operation of Linked Sites. If you access Linked Sites, you do so at your own risk and understand that Wowza accepts no responsibility or liability for the content or operation of Linked Sites. Wowza provides these links only as a convenience, and the inclusion of a link does not imply that Wowza endorses such Linked Sites or any content, products, or services available from Linked Sites.

This document also refers to third-party software that is not licensed, sold, or distributed by Wowza (collectively, "Third-Party Software"). Wowza does not endorse, is not responsible for, and accepts no liability related to Third-Party Software. Please ensure that any and all use of Wowza® software and third-party software is properly licensed.

Wowza Trademarks

Wowza®, Wowza Media Systems®, Wowza Streaming Engine™, along with other trademarks, logos, trade dress, and other proprietary colors and markings, are each trademarks or registered trademarks of Wowza in the United States and in other countries (collectively, "Wowza Marks"). No right to use Wowza Marks in any way is granted hereunder. Contact sales@wowza.com for information on obtaining the right to use Wowza Marks. Any use of Wowza Marks, authorized or otherwise, shall inure to the sole benefit of Wowza.

Third-Party Trademarks and Copyrights

Trademarks, product names, logos, designs, trade dress, and other proprietary markings of non-Wowza third parties (collectively, "Third-Party Marks") may be trademarks or registered trademarks of their respective owners. Use of Third-Party Marks is for the sole purpose of identifying third-party products and services and does not represent endorsement, sponsorship, partnership, or other affiliation between Wowza and such third parties.

A list of applicable copyright notices related to content in this document is available on the Wowza website.

Document History

Version	Description	Release date
4.0	Initial release for Wowza Streaming Engine 4.0	02-21-2014

Table of Contents

Introduction

This document describes how to install and configure Wowza Streaming Engine™ software 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 Streaming Engine that are ready to start using the Amazon Web Services (AWS) Management Console. You can use these AMIs as needed to launch Wowza Streaming Engine for EC2 instances, paying for your running instance time and bandwidth consumption only while the instances are running. You can launch Wowza Streaming Engine 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 Streaming Engine. If you aren't, you can get a free 180-day trial license for Wowza Streaming Engine by completing a request form. The Wowza Streaming Engine Trial Edition download includes the Wowza Streaming Engine software, all premium AddOns, documentation, and examples. The Wowza Streaming Engine 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 Streaming Engine for Amazon EC2.

Getting Started

To get started using Wowza Streaming Engine™ for Amazon EC2 software, 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 Streaming Engine 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 or Perpetual license key with your Wowza Streaming Engine for EC2 instance. When using this option, billing for your running instance time and bandwidth consumption is managed by Amazon. Wowza Streaming Engine Subscription users will continue to receive a separate monthly invoice from Wowza for usage of the Wowza Streaming Engine software and AddOns. This option provides access to all Wowza Streaming Engine 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 Streaming Engine 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 Streaming Engine usage; however, it doesn't provide access to the premium AddOns.

Deploying Wowza Streaming Engine for EC2 Instances

This section describes how to deploy Wowza Streaming Engine™ 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 Streaming Engine 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 of available EC2 regions. Wowza provides public Amazon Linux AMIs that are preconfigured for specific regions. For more information about how to select a region for your Wowza Streaming Engine for EC2 instance, see Choosing a Wowza Streaming Engine AMI.

Choosing a Wowza Streaming Engine AMI

Wowza provides public Amazon Linux AMIs with preconfigured versions of Wowza Streaming Engine that are ready to start through the EC2 Management Console. You must get a current Wowza Streaming Engine AMI ID for each Wowza Streaming Engine 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.

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

Creating a key pair

After you have decided on a Wowza Streaming Engine 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 SSH client 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 Streaming Engine 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.

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 Streaming Engine, see Configuring streaming ports. 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	Custom TCP rule	22
Wowza Streaming Engine Management	Custom TCP rule	8086-8088
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.

Launching the instance

To launch your Wowza Streaming Engine for Amazon EC2 instance, do the following:

1. In the Wowza Prebuilt AMIs webpage, click the link for the Wowza Streaming Engine 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 an 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 an 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 Streaming Engine AMIs are set to use the **m3.medium** instance type by default. To select a different instance type, select the filtering option to show **All instances**, and then click the instance type that you want to use in the list.

Important

If you selected a prebuilt Wowza **DevPay** AMI ID from the Wowza Prebuilt AMIs webpage, the default **m3.medium** instance type that's selected isn't supported and you must choose another supported instance type. Select the filtering option to show **All generations** of instance types, and then choose a supported instance type for your selected EC2 region. You can review the pre-built Amazon DevPay AMIs table in the Wowza Prebuilt AMIs webpage to determine the supported instance types for **DevPay** AMI IDs in the EC2 region that you selected.

Note

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 Details** page enables you to configure optional settings for your selected instance type as well as to configure advanced instance options for loading a Wowza Streaming Engine startup package to launch your Wowza Streaming Engine 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 Streaming Engine 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 Streaming Engine 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
   - vod
   - vods3

Note: Wowza Provides Prebuilt Startup Packages

You can download prebuilt startup packages from the Wowza 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 Streaming Engine 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.

Using the **User data** field, you can specify the startup package and license key(s) to use when starting your instance. The full URL to the startup package and license key data is specified using the comma-delimited key-value pairs. Multiple license keys are separated by the pipe (|) character.

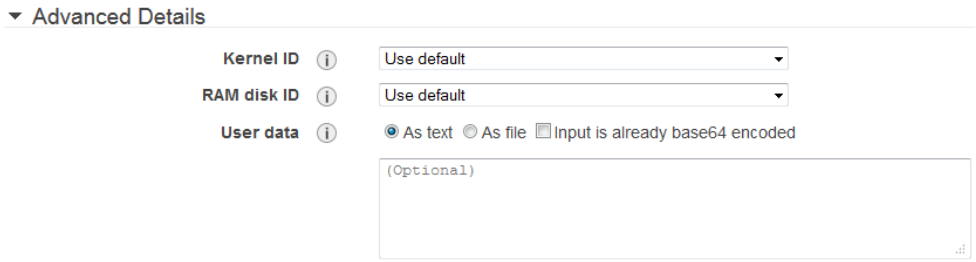
- WZA_startupPackageURL=[url-to-startup-package]
- WZA_wowzaServerLicenseKey=[license-key]|[license-key]

Note

The **WZA_wowzaServerLicenseKey** key-value pair is only needed when using a **LicKey** AMI ID.

To load a Wowza Streaming Engine startup package and specify a Wowza Streaming Engine license key to launch your Wowza Streaming Engine instance with, do the following:

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



2. Enter the key-value pair data into the **User data** field (the example below includes startup package, perpetual and transcoder license keys):

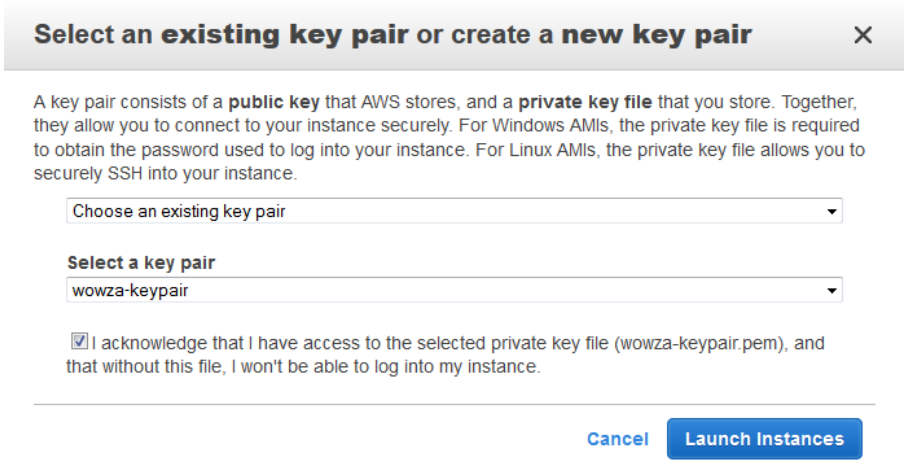
```
WZA_startupPackageURL=
https://s3.amazonaws.com/wowzamediasystems/com/wowza/startup/default
_4.0.0.zip,WZA_wowzaServerLicenseKey=ENGP4-12345-abcde-12345-abcde-
12345|TRN14-12345-abcde-12345-abcde-12345

Pass in multiple license keys for AddOns, such as transcoder and
nDVR, using the pipe (|) character between the keys. For example the
user data might look like this:
WZA_startupPackageURL=http://myserver.com/startuppackage.zip,WZA_wow
zaServerLicenseKey=ENGP4-abcde-HafUT-abcde-A73We-wCCrk|TRN14-abcde-
DKebt-nzzA6-abcde-eGZzX|DRMA4-abcde-PmXdz-EuWw8-abcde-hkTPb
```

Note

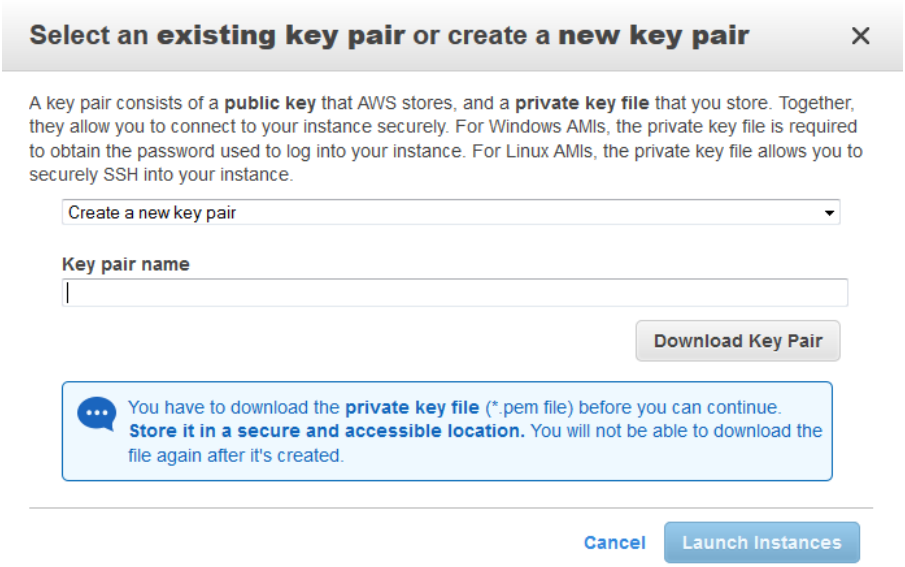
Make sure that the **Input is already base64 encoded** check box is cleared.

3. 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.
4. 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.
5. 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.
6. Click the **Review and Launch** button. On the **Review Instance Launch** 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** check box, and then click the **Launch Instances** button:



To create a new key pair for use with this instance, select **Create a new key pair** in the top list, and then do the following:



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.

7. 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 Streaming Engine 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.

Notes

- If you use a **LicKey** AMI ID and do not specify a **WZA_wowzaServerLicenseKey**, 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 Streaming Engine Subscription 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 public domain name and instance ID of instance

You must use the public domain name (or hostname) and instance ID of your running instance to access the instance remotely for configuration using the Wowza Streaming Engine Manager, to connect to the instance using Secure Shell (SSH) and for streaming. To get the public domain name and instance ID 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 and the **Instance ID** is the instances instance ID.
4. Make note of these values, they are used below.

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 Streaming Engine for EC2 instance is working correctly by logging on to the Wowza Streaming Engine Manager and play your first video using one of the built in test players:

1. Open a web browser and enter the following URL to connect to the Wowza Streaming Engine Manager:
   `http://[public-domain-name]:8088`
2. You should now see the Streaming Engine Manager **Welcome** page. Click the **Skip Intro** button to go directly to the **Sign In** page.
3. Sign with the user name **wowza** and a password. The password is the instance ID you made note of above.
4. In Streaming Engine Manager, click the **Applications** tab in the top navigation bar, and then click the **vod** application in the contents pane.
5. In the upper-right corner of the **vod** application page, click the **Test Players** button. The **Test Players** window that opens includes test players that are preconfigured to stream the **sample.mp4** video file over various streaming formats. Use one of the test players to playback the default **sample.mp4** video file.

The **Big Buck Bunny** video should start to play. You're now ready to configure and use your Wowza Streaming Engine instance on Amazon EC2.

Managing the instance over an SSH session

Most Wowza Streaming Engine Manager configuration can be accomplished using the Wowza Streaming Engine Manager. You may need to open a secure session to your Amazon EC2 instance using Secure Shell (SSH) in order to start and stop the Wowza Streaming Engine or to make server 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 session to your Amazon EC2 instance by using the PuTTY Secure Shell client. 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 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.

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**:
    a. 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 Streaming Engine.
    b. In the **Port** field, enter **22**. Make sure that you have followed the instructions for opening TCP port 22 in Opening ports for streaming.
    c. 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 SSH 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 SSH session.

You should see the **login as:** prompt in the SSH client window. Enter the username **ec2-user** to login to your Amazon EC2 instance. When working with Wowza Streaming Engine, 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 SSH client 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 Streaming Engine 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 Streaming Engine folders are symbolically linked to the **/home/wowza** folder for easy access using FTP.

Wowza Streaming Engine 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 SSH client 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 easiest method to supply a Wowza Streaming Engine license key is to provide it using the **WZA_wowzaServerLicenseKey** key-value pair when the instance is started as outline above (See Launching the instance.). If you do not specify a license key at instance startup, the instance uses a temporary license key. You must replace the temporary license key with a valid Wowza Streaming Engine Subscription 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:

Wowza Streaming Engine Manager

1. Launch your Wowza Streaming Engine for Amazon EC2 instance. (See Launching the instance.)
2. Open a web browser and connect to your instance using the following URL:

   http://[public-domain-name]:8088
3. Login using the user name **wowza** and the password, which is the **Instance ID** of the running Amazon EC2 instance.
4. Click the **Server** link in the top navigation bar to open the server configuration page.
5. Click the **Server Setup** link in the contents pane.
6. Click the **Edit** button and replace the multi-line key in the **License Keys** box with your subscription or perpetual license key and then click the **Save** button.
7. The new license key will take effect after you click the **Restart Now** or **Restart** button on the top right of the page.

SSH client connection

1. Launch your Wowza Streaming Engine for Amazon EC2 instance. (See Launching the instance.)
2. After the instance has started, connect to it using a secure SSH client session. (See Managing the instance over an SSH session.)
3. In the SSH client window, change directory to the **/usr/local/WowzaStreamingEngine/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 or Perpetual license key value, and then save the file.
5. Restart the Wowza Streaming Engine instance to activate the key. You can do this by executing the following commands:

```
sudo service WowzaStreamingEngine stop
sudo service WowzaStreamingEngine 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 Streaming Engine 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.

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 Streaming Engine 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.

Viewing directory contents

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

```
cd /home/wowza
```

From here, you can display the subdirectories in the Wowza Streaming Engine 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 Streaming Engine logs by executing the following commands:

```
cd /usr/local/WowzaStreamingEngine/logs
tail -f wowzastreamingengine_access.log
```

Stopping Wowza Streaming Engine

To stop Wowza Streaming Engine, enter the following command:

```
sudo service WowzaStreamingEngine stop
```

Starting Wowza Streaming Engine

To start Wowza Streaming Engine, enter the following command:

```
sudo service WowzaStreamingEngine start
```

Wowza Streaming Engine Configuration Startup Packages

The previous section described how to launch a generic Wowza Streaming Engine™ for Amazon EC2 instance with most of the example applications installed. You can configure a Wowza Streaming Engine 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.

Wowza provides prebuilt startup packages that you can download from the Wowza 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
```

To see how a startup package is constructed, download and extract the contents of the Wowza Streaming Engine **default** startup package: Download default_startupPackage.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/WowzaStreamingEngine** folder of the running Wowza Streaming Engine 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 Streaming Engine 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 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 Streaming Engine 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_startupPackage.zip

The following application names are configured in the **default** startup package:

- live
- 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.

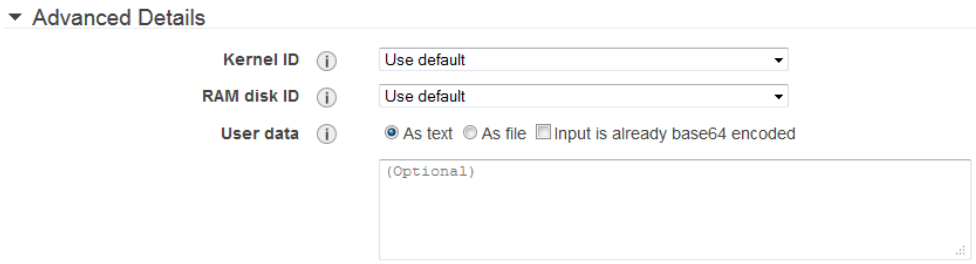
Loading a startup package

When you launch a new instance in EC2 Management Console, you can load a Wowza Streaming Engine startup package to launch the instance with a customized Wowza Streaming Engine configuration. Wowza provides prebuilt startup packages that you can download from the Wowza 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.

There are two methods to load a startup package from the EC2 Management Console: **As text** (key-value pair data references the full URL to startup package) or **As file** (include the startup package contents as **User data**). When using the **As file** method, the startup package must be no more than 16 kb in size.

To load a startup package **As text**, do the following:

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

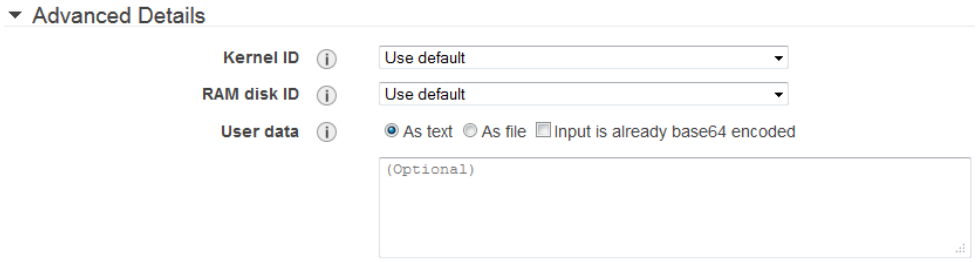


2. Enter the key-value pair **WZA_startupPackageURL=[url-to-startup-package]** in the **User data** field. For example:

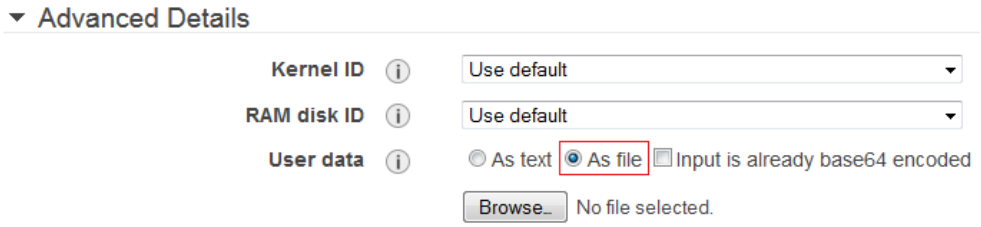
```
WZA_startupPackageURL=
https://s3.amazonaws.com/wowzamediasystems/com/wowza/startup/default_4.0.0.zip
```

To load a startup package **As file**, do the following:

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



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



3. 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** check box is cleared.

- The ability to add startup package user data using the **As file** option 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/WowzaStreamingEngine/logs/wowzastreamingengine_startup.log
```

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

Streaming Media from Amazon S3

You can use Wowza Streaming Engine™ for Amazon EC2 instances to stream media directly from Amazon Simple Storage Service (Amazon S3). Wowza Streaming Engine uses Media Cache caching technology to improve performance when streaming media from Amazon S3. Media Cache 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 Media Cache system is tuned on a per-instance basis in Wowza Streaming Engine startup packages. You can see the tuning in the **tuning/tune.sh** file that's included in any Wowza Streaming Engine startup package.

This feature is included in the **default** startup package for Wowza Streaming Engine 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.m4
v
```

Adobe Flash Player (Flash HDS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4
v/manifest.f4m
```

Apple iOS device (Apple HLS)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4
v/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4
v/Manifest
```

MPEG-DASH player

```
http://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4
v/manifest.mpd
```

RTSP/RTP player or device

```
rtsp://[instance-public-
dns]/vods3/_definst_/mp4:amazons3/mybucket/videos/coolvideos/mycoolvideo.m4
v
```

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_**. Also note that you must use the literal 'amazons3' in the path, as shown in the examples above, not the equivalent segment of the link specified by the properties page of your S3 bucket.

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, modify the Media Cache configuration using the Wowza Streaming Engine Manager:

1. Open a web browser and enter the following URL to connect to the Wowza Streaming Engine Manager and login:

   http://[public-domain-name]:8088
2. Click on the **Server** link in the top navigation bar.
3. Click on the **Media Cache** link in the left navigation pane.
4. Click on the **Sources** tab and select the pencil icon for **amazons3** source item.
5. Enter your **AWS Access Key ID** and **AWS Secret Access Key** and click the Save button.

You will now be able to re-stream protected contenting using the **vods3** application.

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/manifes
t.f4m
```

Apple iOS device (Apple HLS)

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

Microsoft Silverlight (Smooth Streaming)

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

MPEG-DASH player

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

RTSP/RTP player or device

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

Wowza Streaming Engine Instance

A Wowza Streaming Engine™ 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 Streaming Engine
- 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 Streaming Engine for Amazon EC2 AMIs.

Wowza Streaming Engine details

Wowza Streaming Engine is installed at its default location:

```
/usr/local/WowzaStreamingEngine
```

The **Wowza Streaming Engine 4.0.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 Streaming Engine is managed using the following ports:

```
8084                - JMX/JConsole Management
8085                - JMX/JConsole Management
8086                - Stream Manager and Administration
8087                - REST API used by Wowza Streaming Engine Manager
8088                - Streaming Engine Manager
```

```
21                  - FTP access
22                  - SSH access
```

The **Wowza Streaming Engine Manager** runs on TCP port 8088.

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/WowzaStreamingEngine/conf/jmxremote.access
/usr/local/WowzaStreamingEngine/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");
```

The available properties are:

```
- Amazon instance id
com.wowza.amazonaws.ec2.AWSEC2_METADATA_INSTANCE_ID

- Security group
com.wowza.amazonaws.ec2.AWSEC2_METADATA_SECURITY_GROUPS

- Local IP address
com.wowza.amazonaws.ec2.AWSEC2_METADATA_LOCAL_IPV4

- Launch index
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_LAUNCH_INDEX

- Public host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_HOSTNAME

- DevPay product code
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PRODUCT_CODES

- Instance type (ex. m1-small)
com.wowza.amazonaws.ec2.AWSEC2_METADATA_INSTANCE_TYPE

- Public host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_HOSTNAME

- Local host name
com.wowza.amazonaws.ec2.AWSEC2_METADATA_LOCAL_HOSTNAME

- Public IP address
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_IPV4

- S3 manifest path
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_MANIFEST_PATH

- Instance reservation ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_RESERVATION_ID

- AMI ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AMI_ID

- List of ancestor IDs
com.wowza.amazonaws.ec2.AWSEC2_METADATA_ANCESTOR_AMI_IDS

- Kernel ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_KERNEL_ID
```

```
- Availability zone
com.wowza.amazonaws.ec2.AWSEC2_METADATA_AVAILABILITY_ZONE

- Public keys
com.wowza.amazonaws.ec2.AWSEC2_METADATA_PUBLIC_KEYS

- RAM disk ID
com.wowza.amazonaws.ec2.AWSEC2_METADATA_RAMDISK_ID
```

Note

To learn more about Wowza Streaming Engine module development, download and install the Wowza Eclipse software update.

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 Streaming Engine™ software installation folder **/usr/local/WowzaStreamingEngine**.

```
<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 Streaming Engine installation folder. The folder's directory structure should be the same as the Wowza Streaming Engine 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
```

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/WowzaStreamingEngine** folder of the running Wowza Streaming Engine 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**.

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**.

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]
                  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 Streaming Engine **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 file path, 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 Streaming Engine installation folder **/usr/local/WowzaStreamingEngine**.

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 file path, 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 (ex. m1-small)
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
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
```

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