Azure Video on Demand (VOD) Solution: Wowza Streaming Engine, Aspera Faspex and Azure Media Services

Hands on Lab Manual

This hands-on lab session will give you a brief introduction of Video on Demand solution stack on Azure built with services from Aspera, Wowza and Azure Media Services.

I. Introduction

Welcome to the Launch and Learn hands on lab session at Ignite. In this session, you will be able to understand how a fully integrated Video on Demand (VoD) stack is put together on Azure and use the solution tack to take a video file through the workflow.

The Quickstart solution is comprised of Aspera's Faspex file transfer service, Wowza's media streaming engine and Azure Media Services all working together to provide a Video on Demand environment. Quickstart templates make it very simple and fast for enterprises to pilot (and validate) fully baked solutions before putting into production. The hands-on lab will demonstrate how:

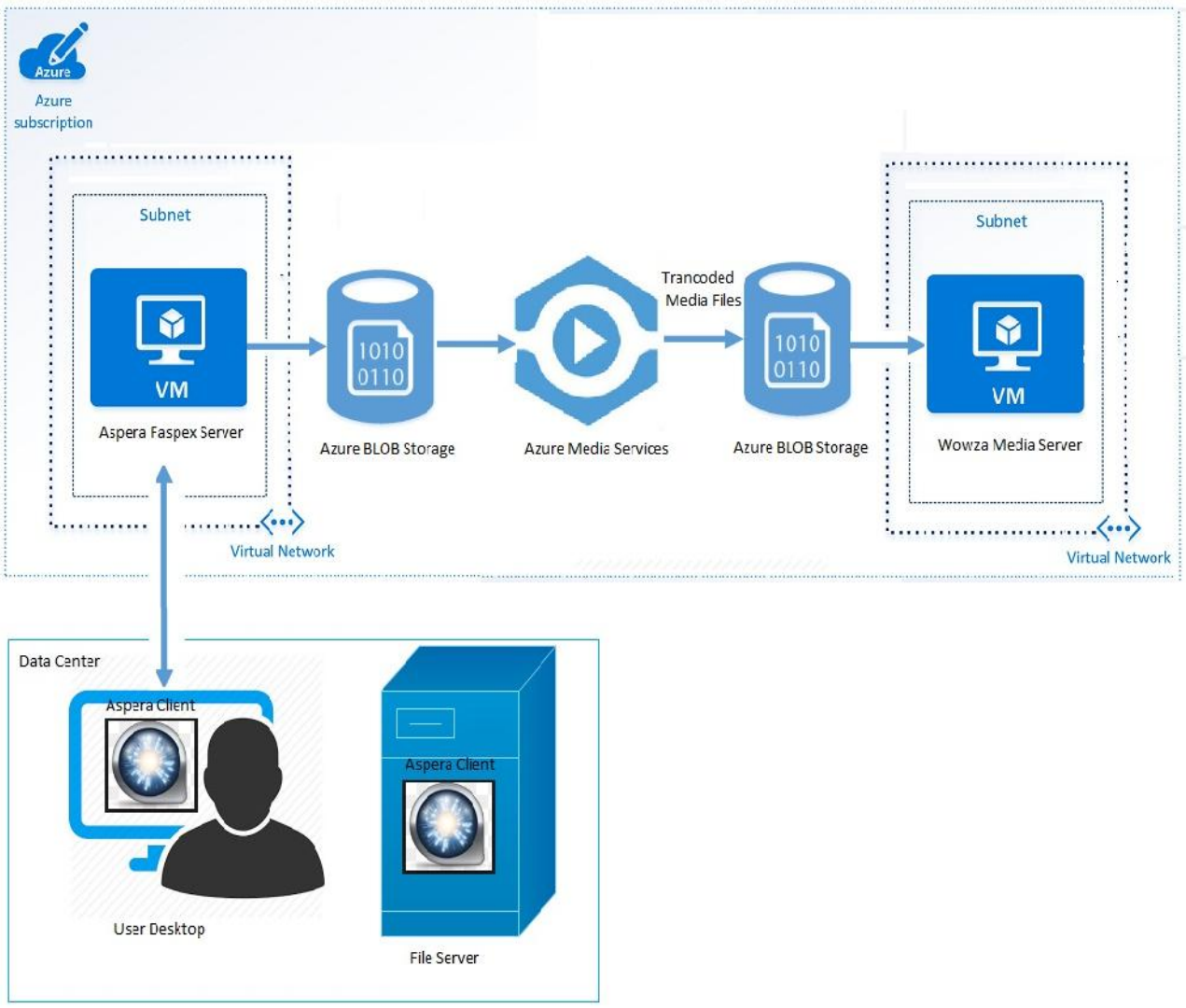
- Aspera Faspex, deployed in Azure is used to move the media files from the client machine to the Azure storage.
- Azure Media Services is used to run transcoding jobs on the inputted media files and output them to azure storage.
- And, Wowza streaming engine to pick up the transcoded media files and deliver/stream them to the clients.

The deployment of the Quickstart solution takes around 15 to 20 minutes' start to finish. For the purposes of this lab, the stack is already pre-deployed.

II. What you will learn in this lab session?

1. How the arm templates are integrated with Aspera, Azure Media services and Wowza Streaming Engine.
2. Transfer files to Azure storage using Aspera's Faspex client and Server.
3. Transcoding of video file(s) with Azure Media Services API
4. Streaming video content through Wowza Streaming Engine.

III. Solution Overview



Components of the Quickstart Solution

The diagram above shows the overall deployment architecture for the Video on Demand solution stack. As a part of deployment, the template launches the following:

- Aspera Faspex Server
- Azure Storage Accounts
- Azure Media Services
- Wowza Streaming Server

Aspera Faspex Server

Aspera Faspex will be running in a Linux Centos Machine. The size of VM is Standard DS1. Aspera's transfer service can move terabytes of data in and out of Azure Blobs, as well as local storage, up to 100x faster than FTP. The Aspera application platform supports a variety of Aspera or custom client options for desktops, web and mobile transfers. For more information on Aspera, please visit https://www.aspera.com

Azure Storage Account

"End Users" will transfer the video files from their desktop to an Azure storage account "through" the Aspera connect client and are saved inside Azure Blob's storage container(s). For more information on Azure Blobs, please visit https://azure.microsoft.com/en-us/

Azure Media Services

Azure Media Services is used for encoding/transcoding content for delivery across multiple formats and devices. In this lab user will transcode to multiple bitrate MP4 files and deliver them dynamically to the latest adaptive bitrate streaming protocols. For more information please visit https://azure.microsoft.com/en-us/services/media-services/

Wowza Streaming Engine Server

Wowza Streaming Engine server is used for streaming of on-demand video over IP networks to desktops, laptops, tablets and mobile devices. Wowza Streaming Engine can stream to multiple types of playback clients and devices simultaneously, including the Adobe Flash player, Microsoft Silverlight player and Apple QuickTime Player. For more information, please visit https://www.wowza.com/

IV. Tools needed for the hands-on lab

All the tools needed to complete the lab are pre-installed on the jump-host VM.

1. Azure Portal with access will be **provided by the instructor**
    - Username: ****@2016ignite.onmicrosoft.com
        - Actual user id will be provided in the class.
    - Password: Instructor will provide the password.
    - Azure Portal access at : https://portal.azure.com

V. Hands on lab steps

Access and key information:

The following information is needed to connect to each server. Please keep referring to them as needed.

- Customer ID: 89280462-5bf1-4371-95cc-ab4cb2f33a5e
- Entitlement key: cc924817-0621-457e-95ce-8dcb6813b648

- **Desktop login: duser/Admin@123**
  Provided as parameters while deploying and displayed in output

- Aspera server ssh login: auser/Admin@123
  Provided as parameters while deploying and displayed in output

- Wowza server ssh login: wuser/Admin@123
  Provided as parameters while deploying and displayed in output

- Wowza web portal login: wowza/Ignite@2016
  Deployed as part of the ARM script.

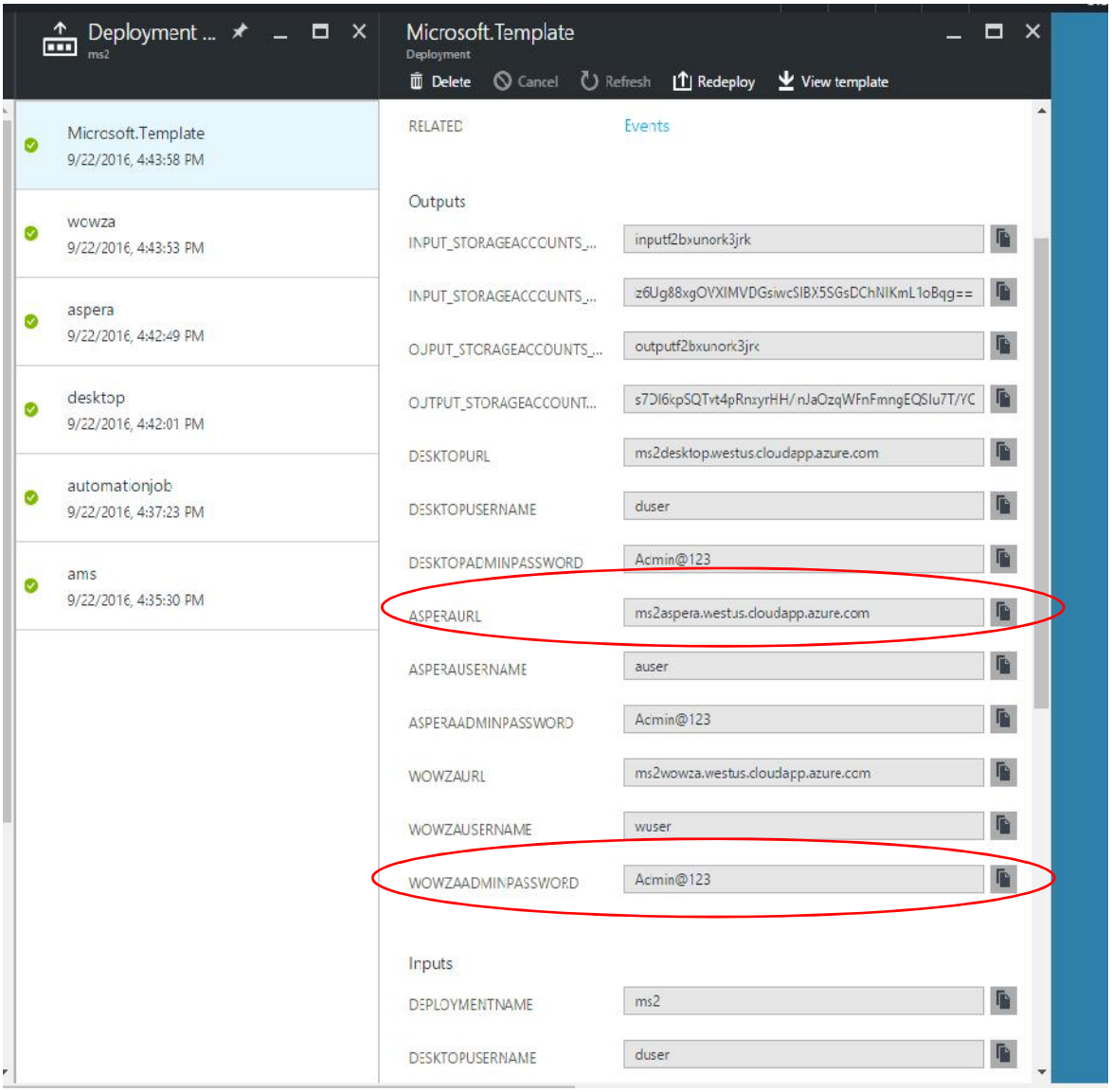
1. Access the azure portal

Your Azure Video on Demand stack/instance is already deployed for you.

Please log into the Azure Portal at **https://portal.azure.com** with the user id/password provided.

Access the portal and review the resources deployed.

- Click on the resource group.
- Click the last deployment and review the outputs.

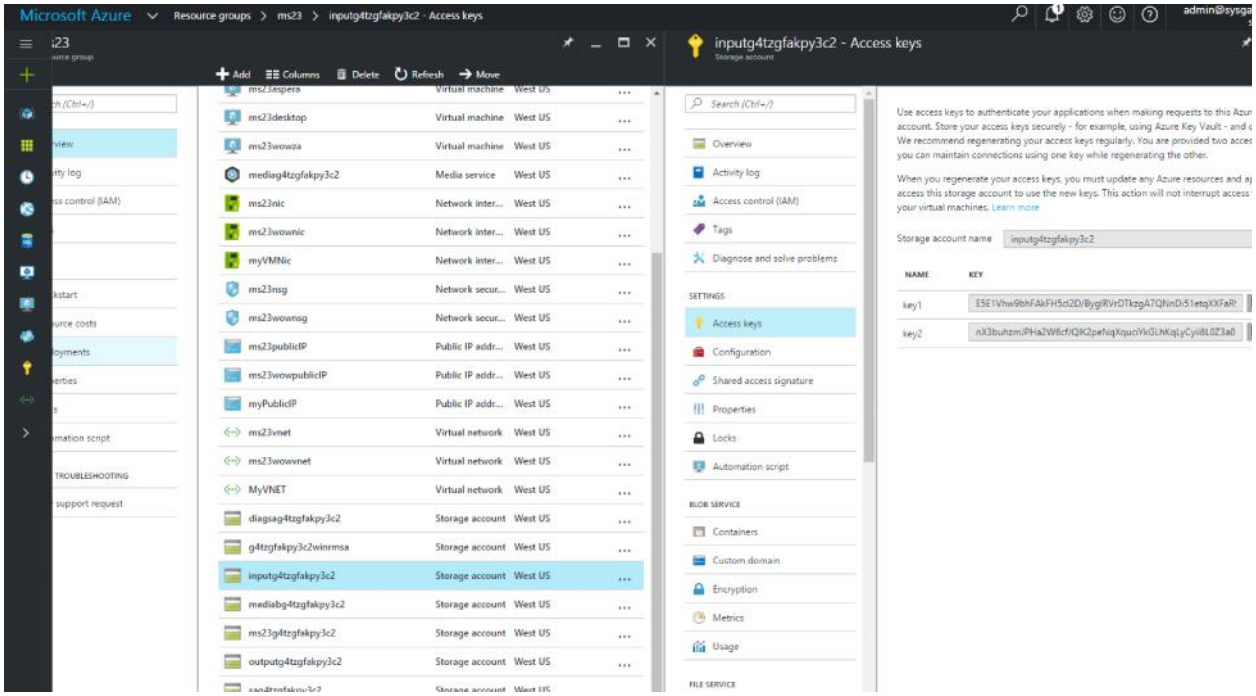


- Note down the Server name/IP name for **desktop**, aspera and wowza VMs which are deployed. They will have the name of *aspera and *wowza respectively.
    - **Example**: ams21aspera.centralus.cloudapp.azure.com and ams21wowza.centralus.cloudapp.azure.com
    - **Note the actual server names from your deployment from the Azure portal.**

- Note down the input storage account name and their access key. The input storage account will have a name of input* under storage account type. Click on the name and go to Settings/Access keys to get the key 1. Review the screenshot below to how to get to storage account and access keys.

    - **Example**: Storage account name: inputyizo6zxgqmgd2 Storage account access key1: RjbLVcWfbLWDxRR/muEVQsKhTCs07/Ic2k0K3QxnyrQ/OM1BlpUEHX+At6FnIBuia3mPI9r2Evd1PbOKN80lyg==
    - **Note: Please make sure you get the actual values from your deployment using the azure portal**

We need both information to move the data from the client to the input storage.



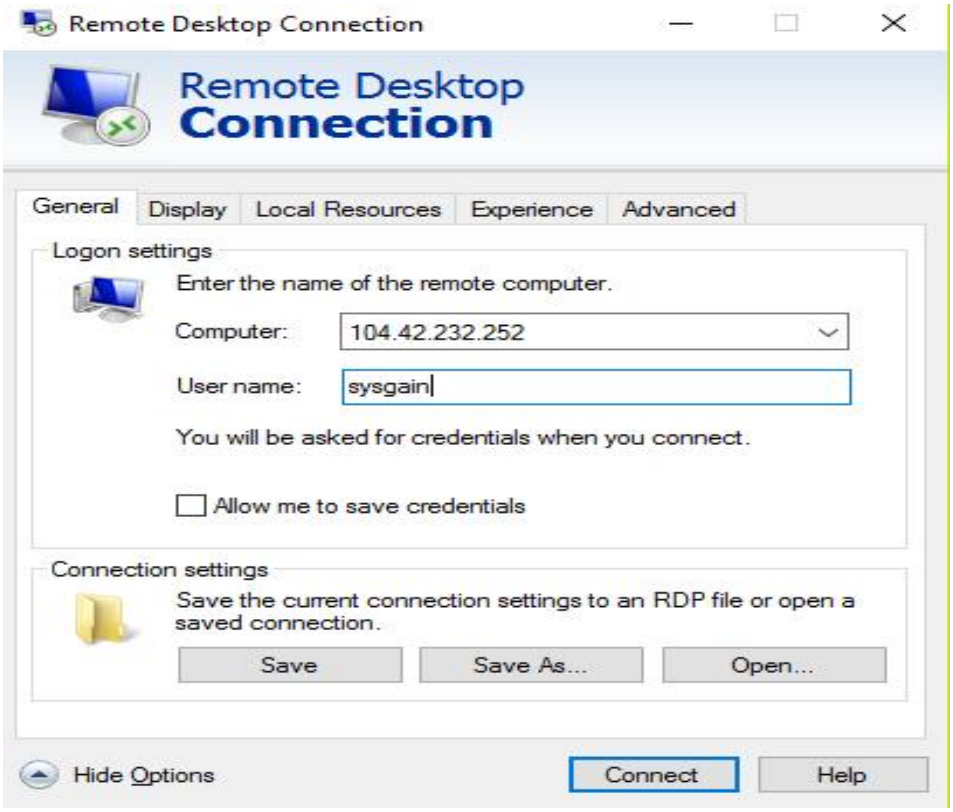
2. Remote Desktop into the Jump Host

The jump host is already configured with all necessary software/tools. Open remote desktop and connect to the windows server. Please review the following software shortcuts on desktop.

- Putty ssh client
- Aspera client
- Azure storage explorer

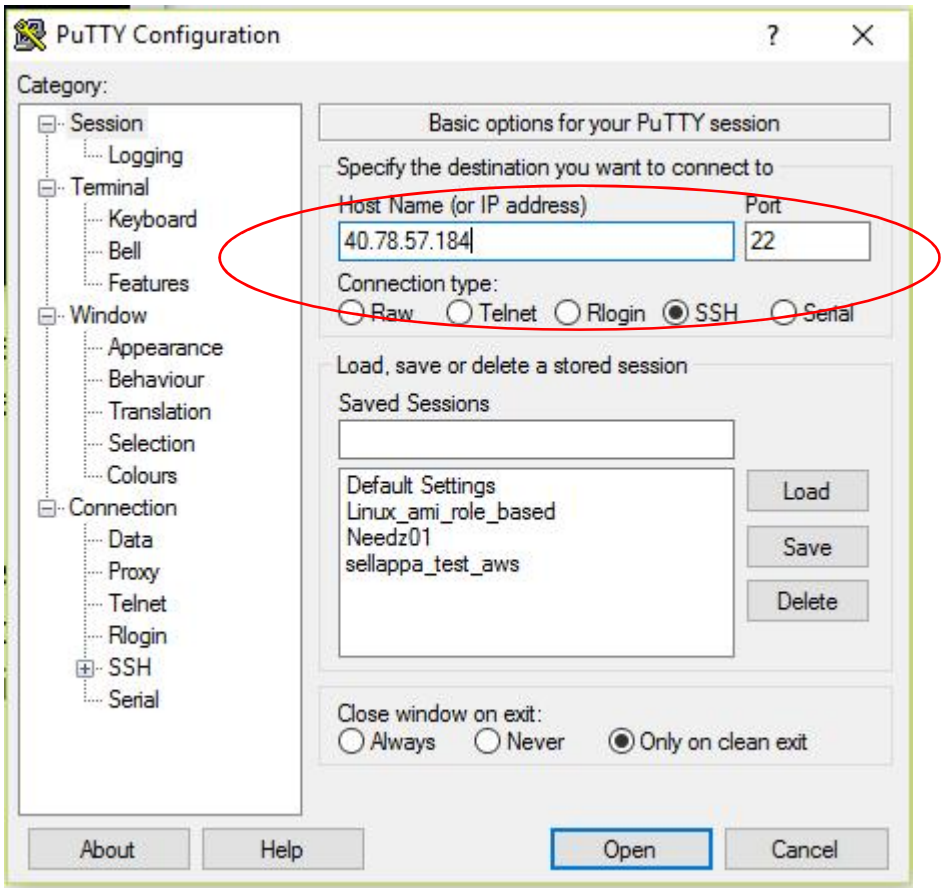
From the location machine search for Remote Desktop Connection, select the application and click. It will show a popup as below.

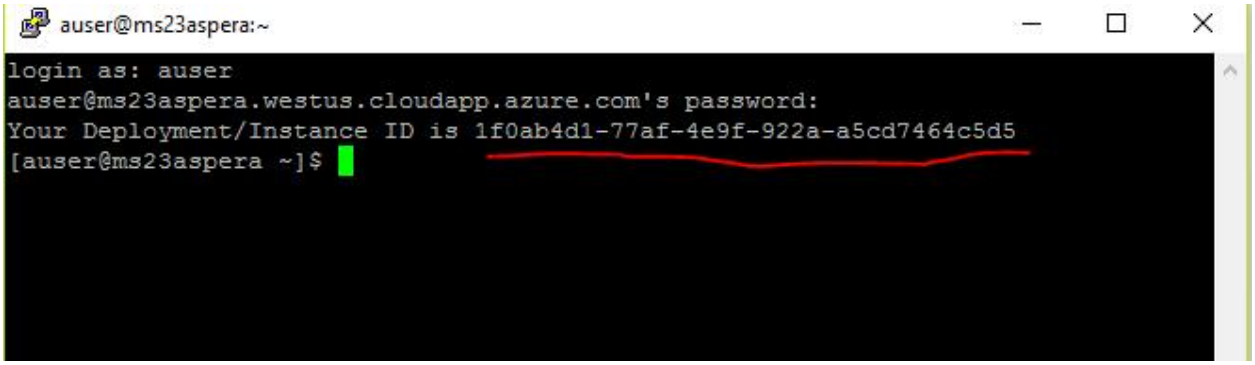
Provide the computer IP address and username/password as provided by the instructor.



3. Access Aspera Faspex Server using SSH

Open the putty application and open an ssh connection to the aspera server. On prompt accept and provide login information which you collected from the output window.

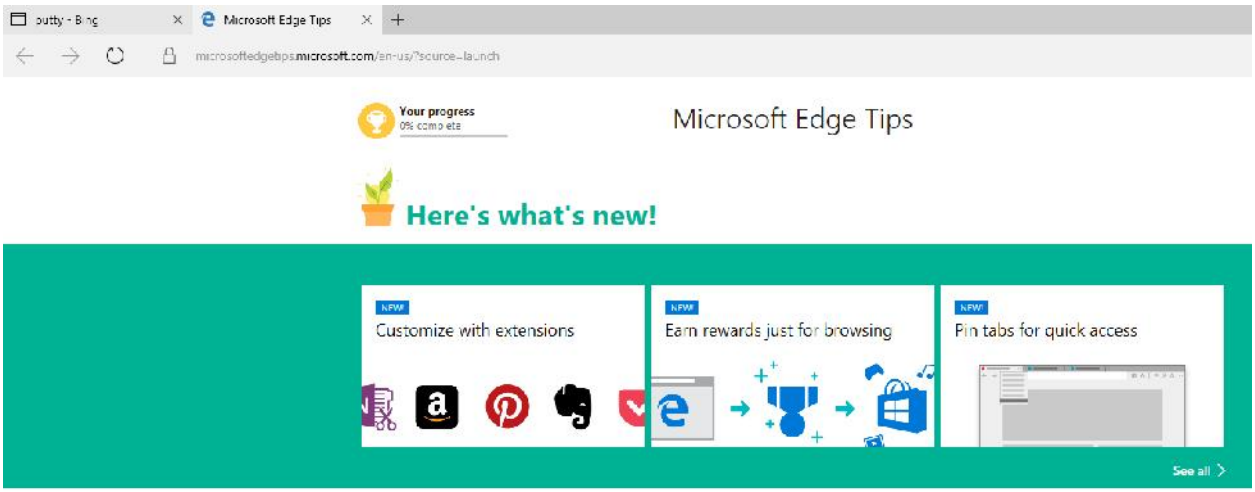




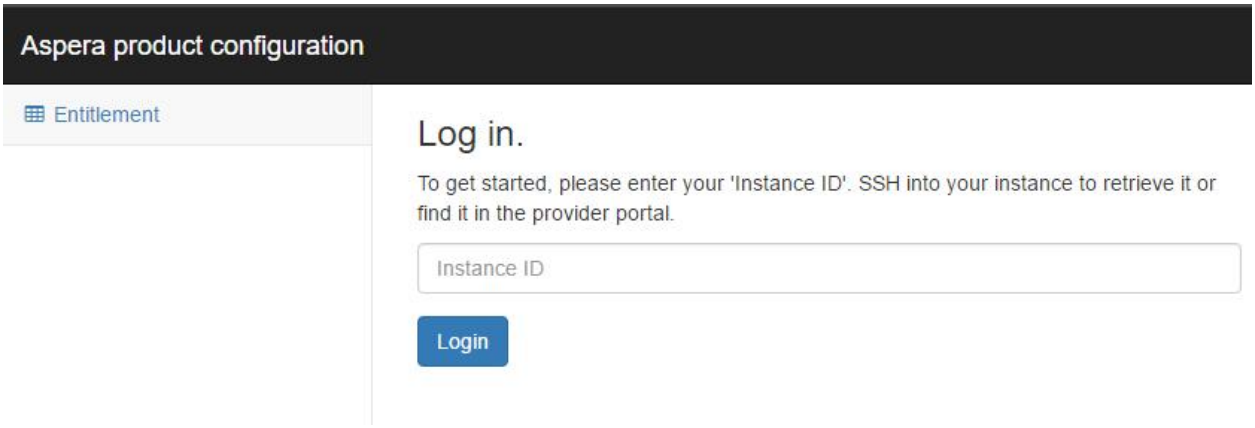
Note down the instance id by double clicking on it. It will be copied to the clipboard.

4. Access Aspera web app to configure

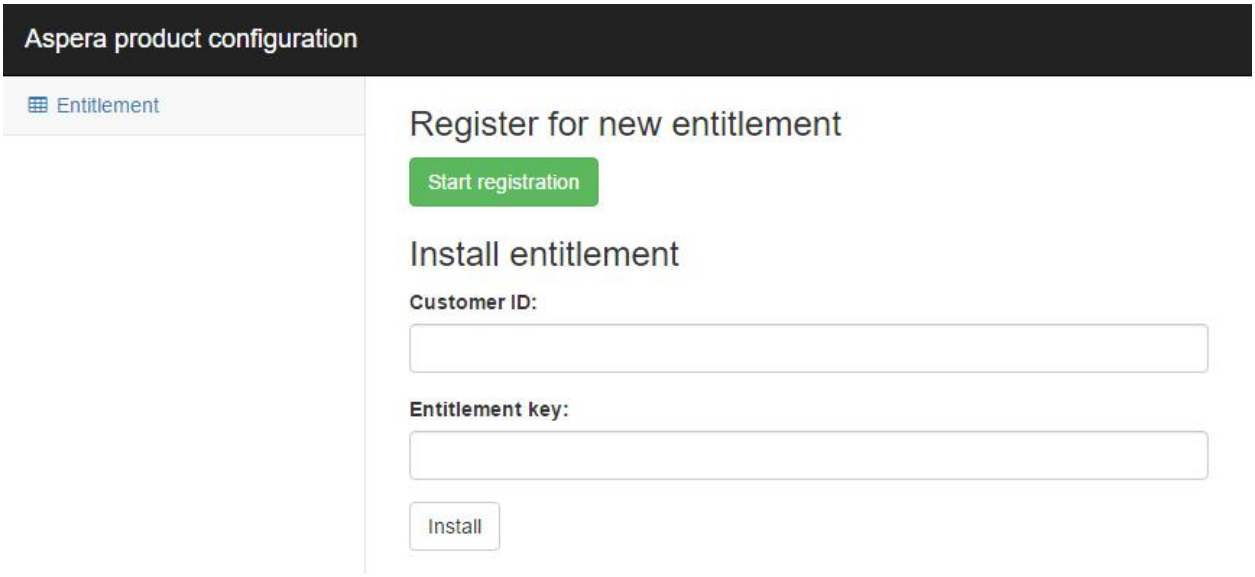
- Open a web browser from the desktop



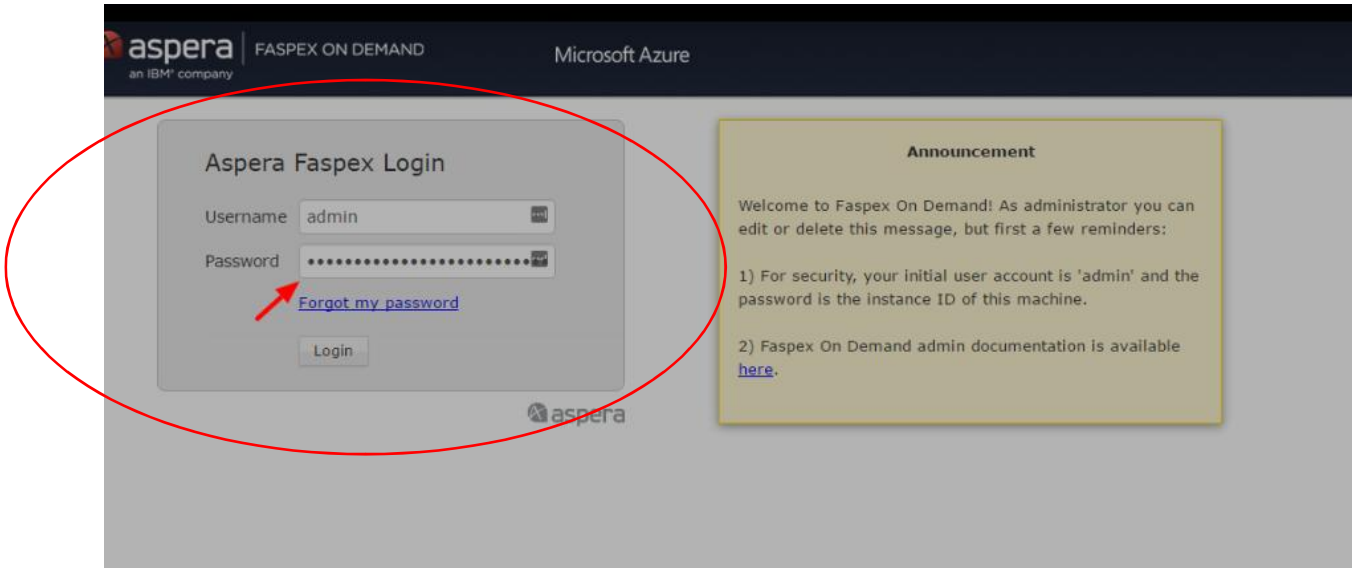
- Login to the Aspera web application by pointing the browser to https://<asperaServerName>/setup
- Enter instance Id you copied from the previous step and submit. See below for screenshot.



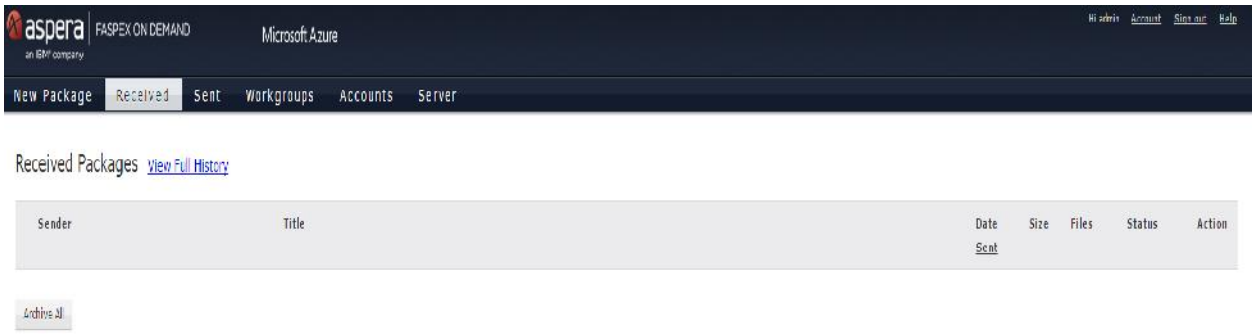
- The interface would ask for customer id and entitlement id.
- Enter the following info and click “install” button.
    - Customer ID: 89280462-5bf1-4371-95cc-ab4cb2f33a5e
    - Entitlement key: cc924817-0621-457e-95ce-8dcb6813b648



- After a few seconds you will be prompted with a login screen
- Enter “admin” for username and the instance id from the previous step for password. On login change the password to “Admin@123”

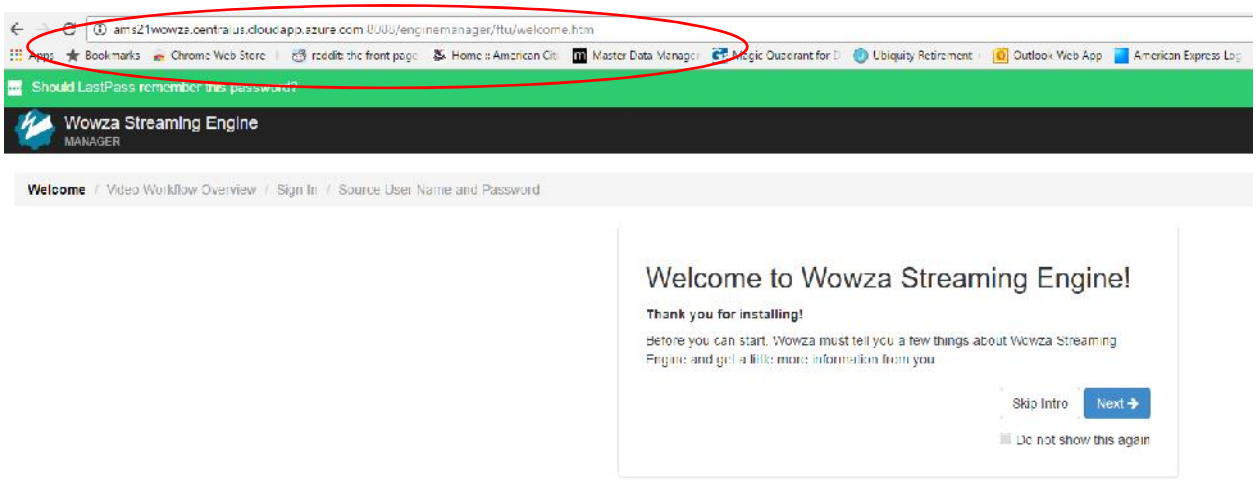


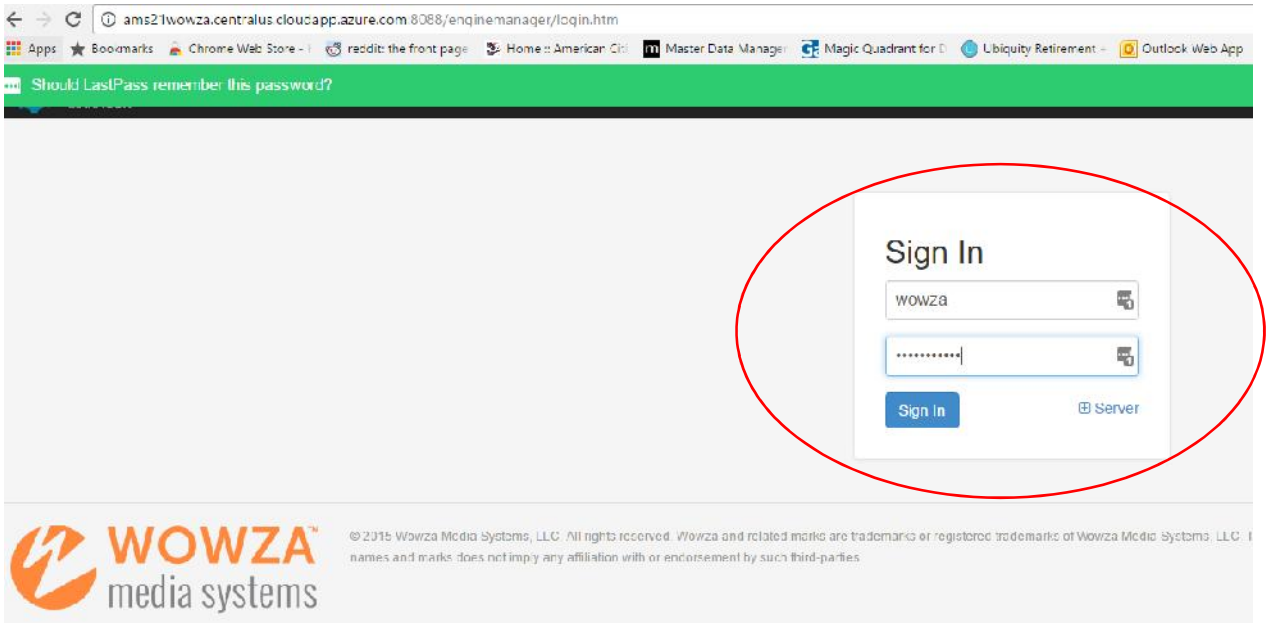
After password change, you will be prompted with a login screen. Upon logging in with Admin and new password that you gave in the previous step (Admin@123), the below portal home page is displayed. This completes the setup of Aspera Server.



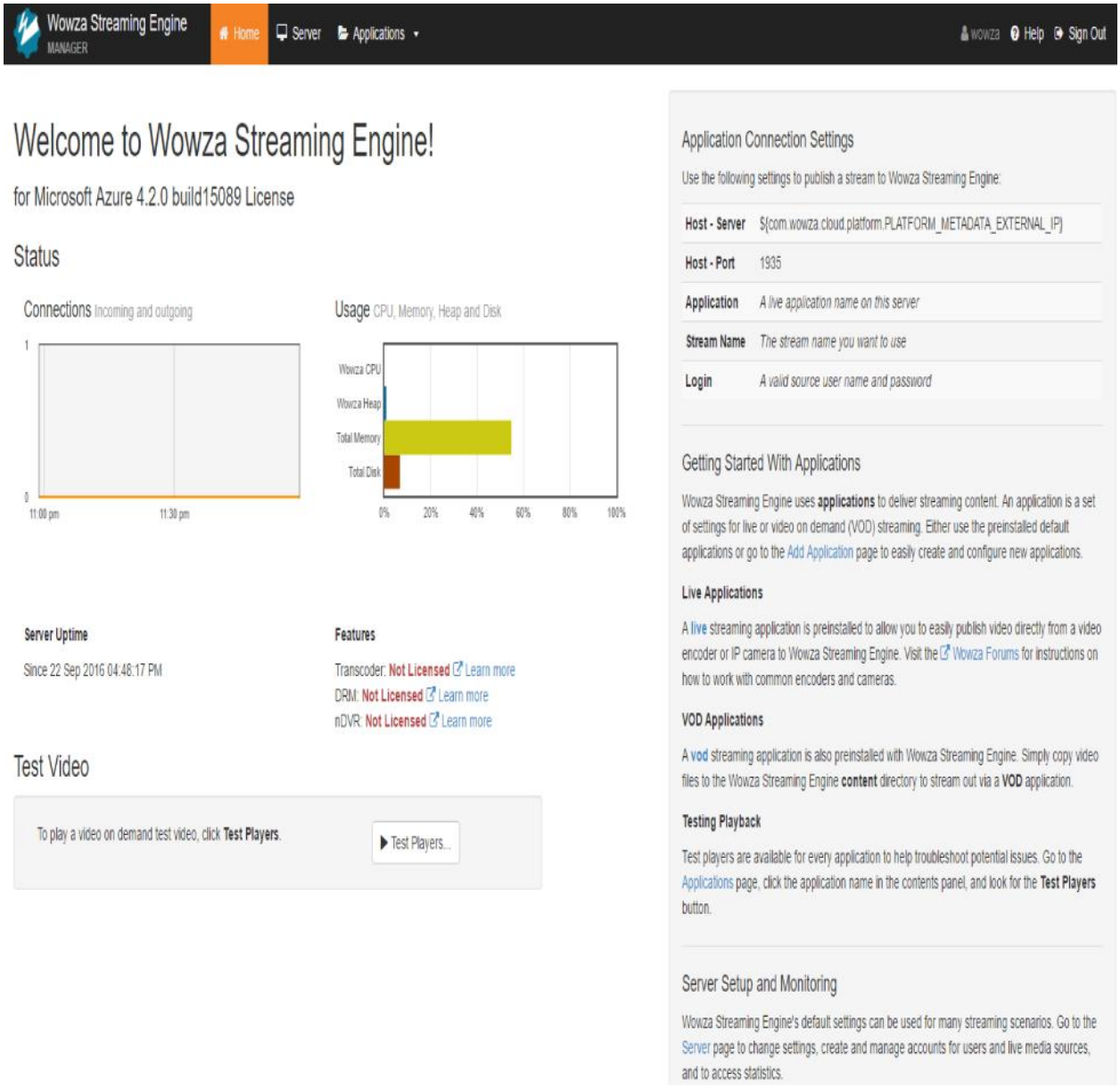
5. Connect to wowza Streaming engine

- Login to the wowza web application by opening the browser to http://<wowzaServerName>:8088/enginemanager
- Login with username and password **wowza/Ignite@2016**



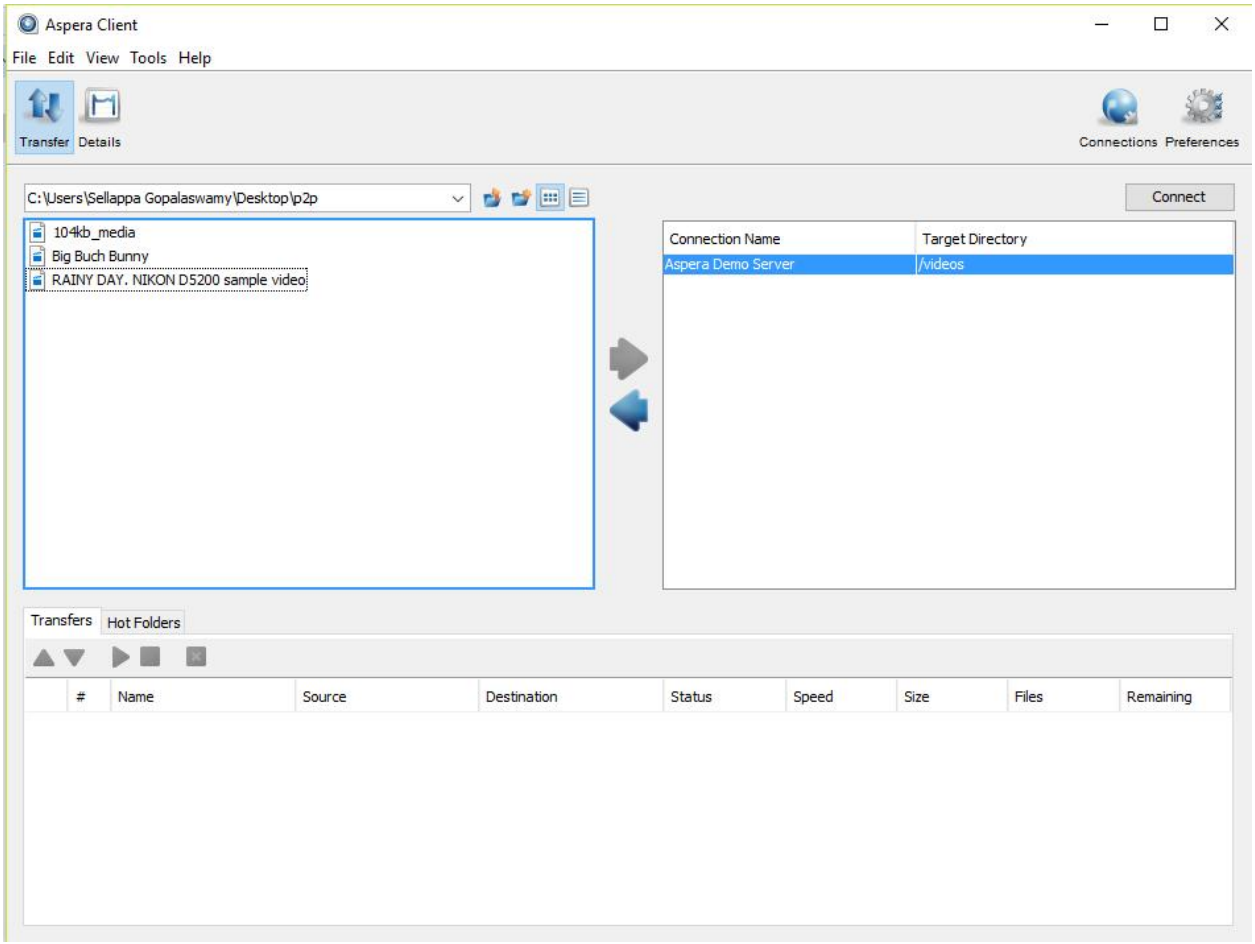


- Once logged in, you will be presented with the welcome screen shown below.


Wowza Streaming Engine
MANAGER
Home
Server
Applications
wowza
Help
Sign Out
Welcome to Wowza Streaming Engine!
for Microsoft Azure 4.2.0 build15089 License
Status
Connections Incoming and outgoing
1
0
11:00 pm
11:30 pm
Usage CPU, Memory, Heap and Disk
Wowza CPU
Wowza Heap
Total Memory
Total Disk
0%
20%
40%
60%
80%
100%
Server Uptime
Since 22 Sep 2016 04:48:17 PM
Features
Transcoder: Not Licensed Learn more
DRM: Not Licensed Learn more
nDVR: Not Licensed Learn more
Test Video
To play a video on demand test video, click Test Players.
Test Players...
Application Connection Settings
Use the following settings to publish a stream to Wowza Streaming Engine:
Host - Server ${com.wowza.cloud.platform.PLATFORM_METADATA_EXTERNAL_IP}
Host - Port 1935
Application A live application name on this server
Stream Name The stream name you want to use
Login A valid source user name and password
Getting Started With Applications
Wowza Streaming Engine uses applications to deliver streaming content. An application is a set of settings for live or video on demand (VOD) streaming. Either use the preinstalled default applications or go to the Add Application page to easily create and configure new applications.
Live Applications
A live streaming application is preinstalled to allow you to easily publish video directly from a video encoder or IP camera to Wowza Streaming Engine. Visit the Wowza Forums for instructions on how to work with common encoders and cameras.
VOD Applications
A vod streaming application is also preinstalled with Wowza Streaming Engine. Simply copy video files to the Wowza Streaming Engine content directory to stream out via a VOD application.
Testing Playback
Test players are available for every application to help troubleshoot potential issues. Go to the Applications page, click the application name in the contents panel, and look for the Test Players button.
Server Setup and Monitoring
Wowza Streaming Engine's default settings can be used for many streaming scenarios. Go to the Server page to change settings, create and manage accounts for users and live media sources, and to access statistics.

6. Connect to Aspera Faspex server using the Aspera client

Choose the remote desktop session and open the Aspera client installed on the jump host. You will find the Aspera client icon on the Desktop.

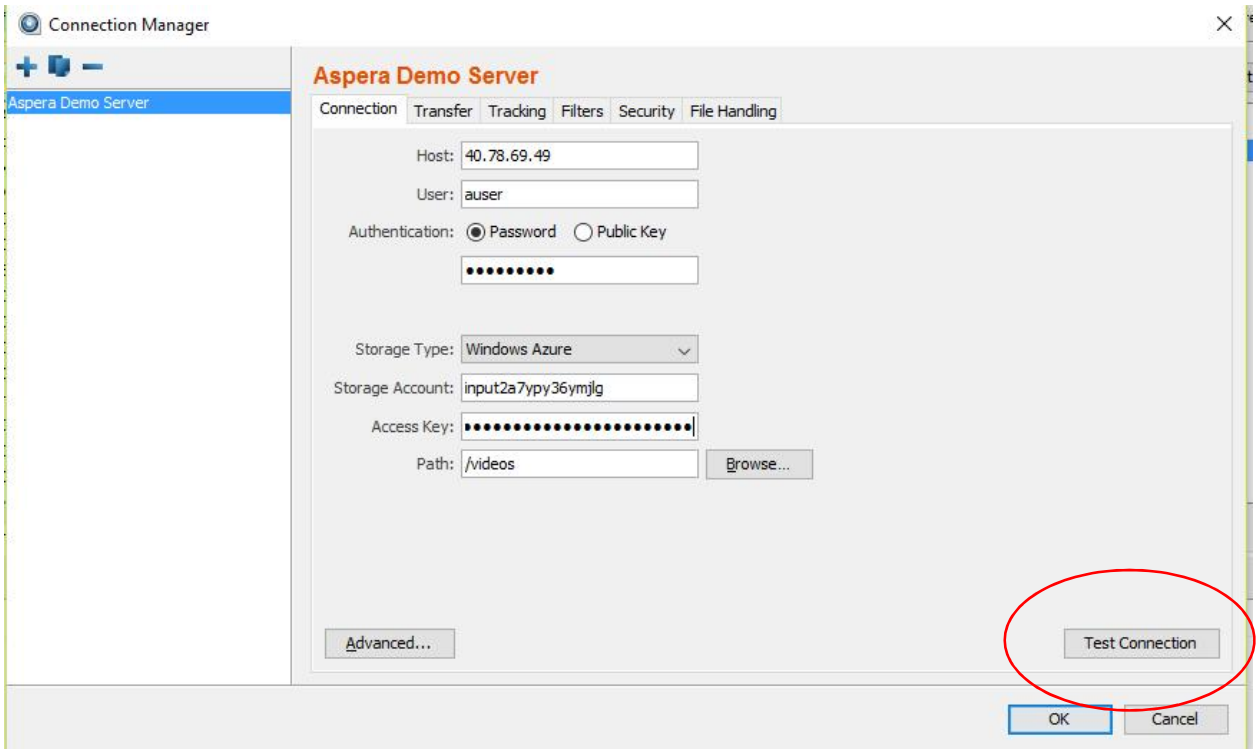


To setup Aspera client to connect to the Aspera Server, collect the Aspera server name, user id and password and azure storage details.

- Aspera Server name collected from output or azure portal.
- User id: auser

- Password : Admin@123
- Storage account and access key from the first step

The media file will land in the above storage account.



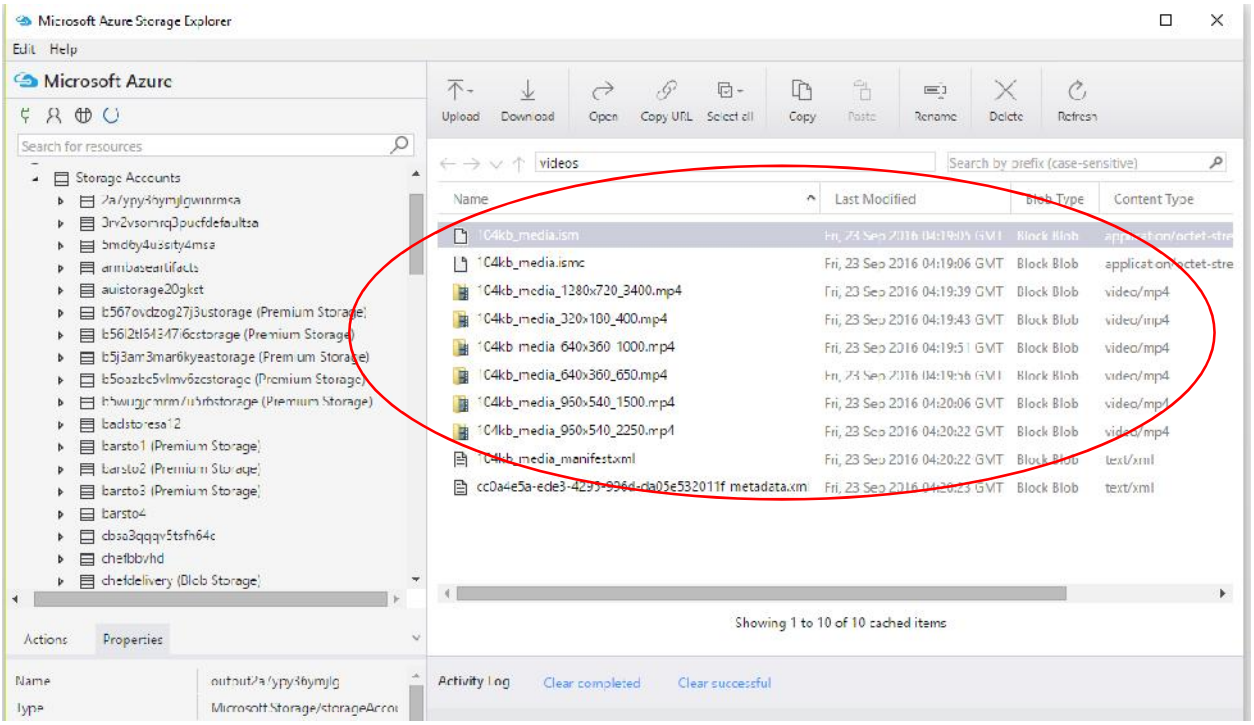
Fill in all the information and hit “Test Connection” to receive a success message.

Note: Storage account here is the input storage account which starts as input* that you have seen in the earlier section.

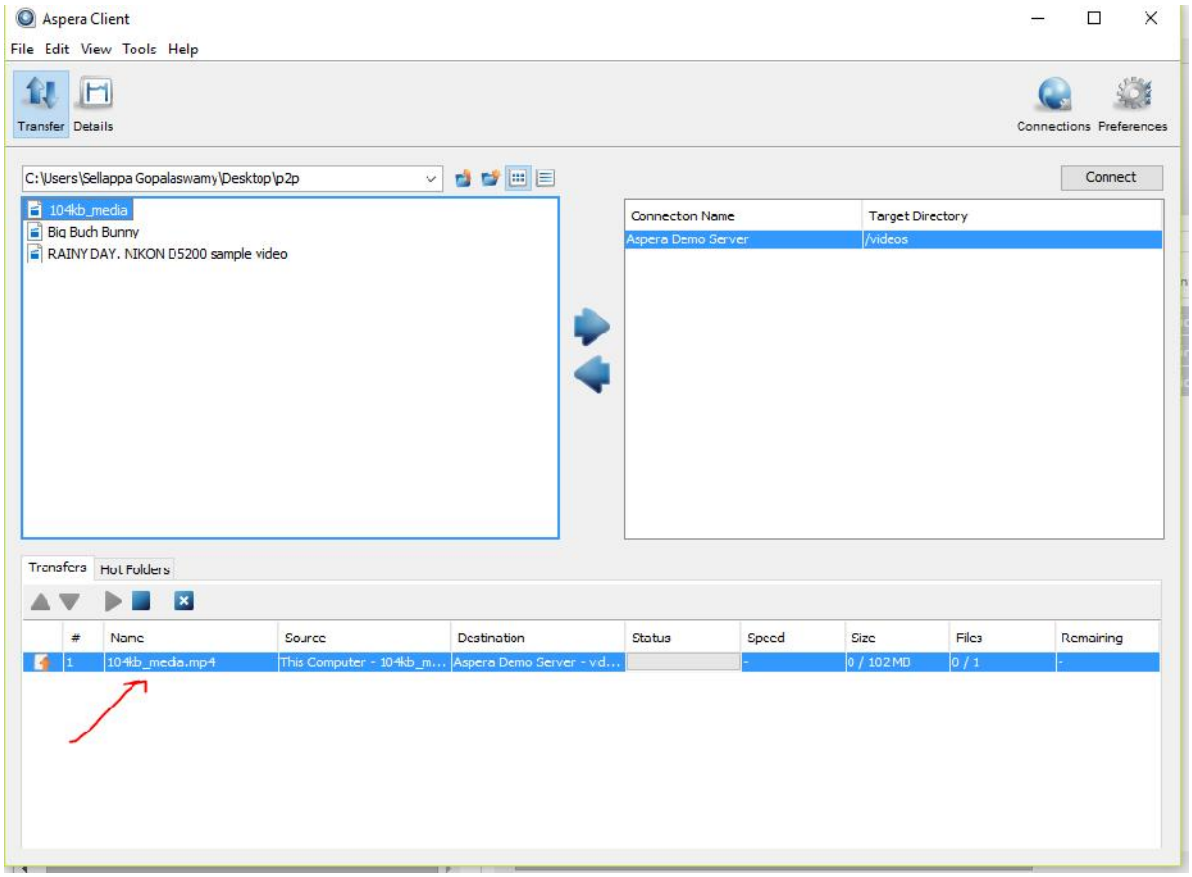
7. Open and connect to Azure storage using Storage Explorer

From the desktop in jump host open the Azure storage explorer and provide the Azure credentials. Browse to Azure storage accounts and view the input and output storage accounts which are deployed as part of your deployment.

This step will show you where and how the azure blobs are organized and how the media files are moving through the infrastructure.



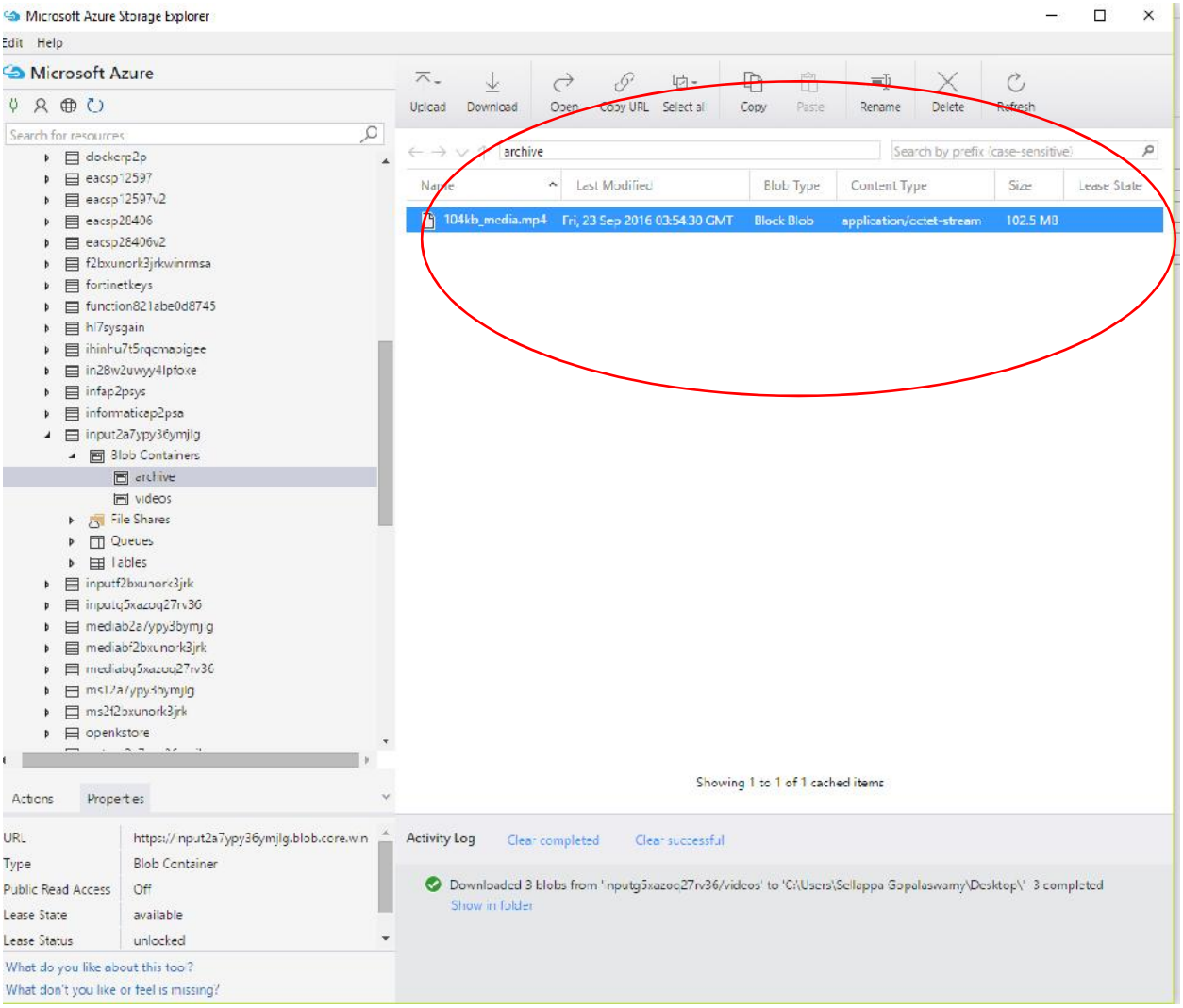
8. Upload a video using the Aspera client



From the left panel, select desktop and select the mp4 file and then press right arrow to transfer the file to the Aspera server. The status of the transfer can be seen in the bottom window.

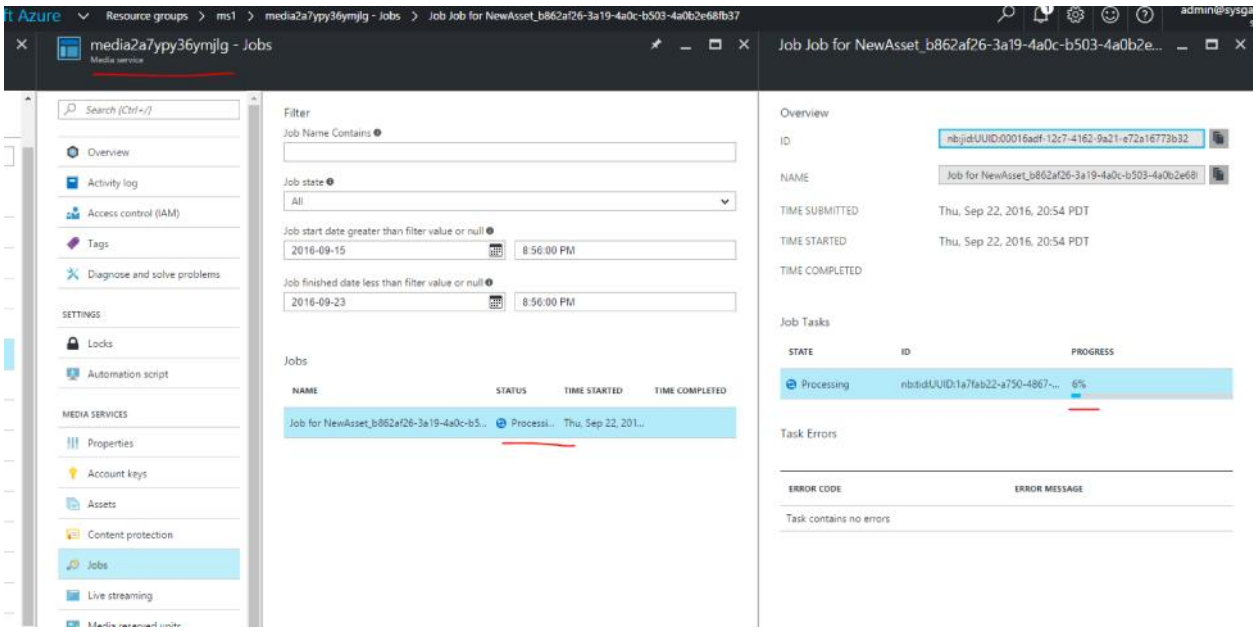
9. Review media file(s) are in Azure Blobs

Check the input storage account and you will see the file transferred to the Azure Blob storage. Aspera will transfer large amounts of data into Azure very quickly, which is a typical use-case scenario in enterprises.



10. Review Azure media services job

The server job deployed as part of the solution stack and running in the background will up the file from input blob, create assets and run "**H264 Multiple Bitrate 720p**" transcoding process which will generate medial files in different formats.

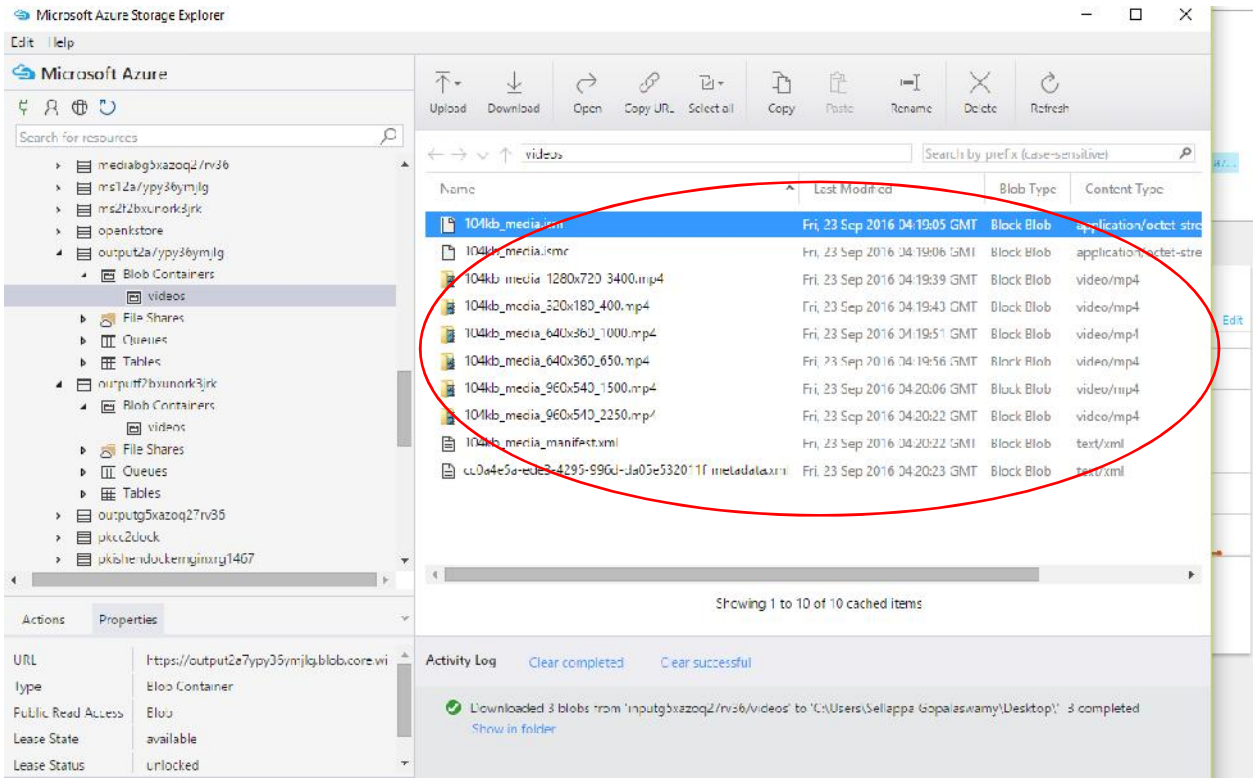


Once completed, check the output Blob using the Azure Storage Explorer as shown below.

Example outputs:

nature_1280x720_3400.mp4
nature_320x180_400.mp4
nature_640x360_1000.mp4
nature_640x360_650.mp4
nature_960x540_1500.mp4

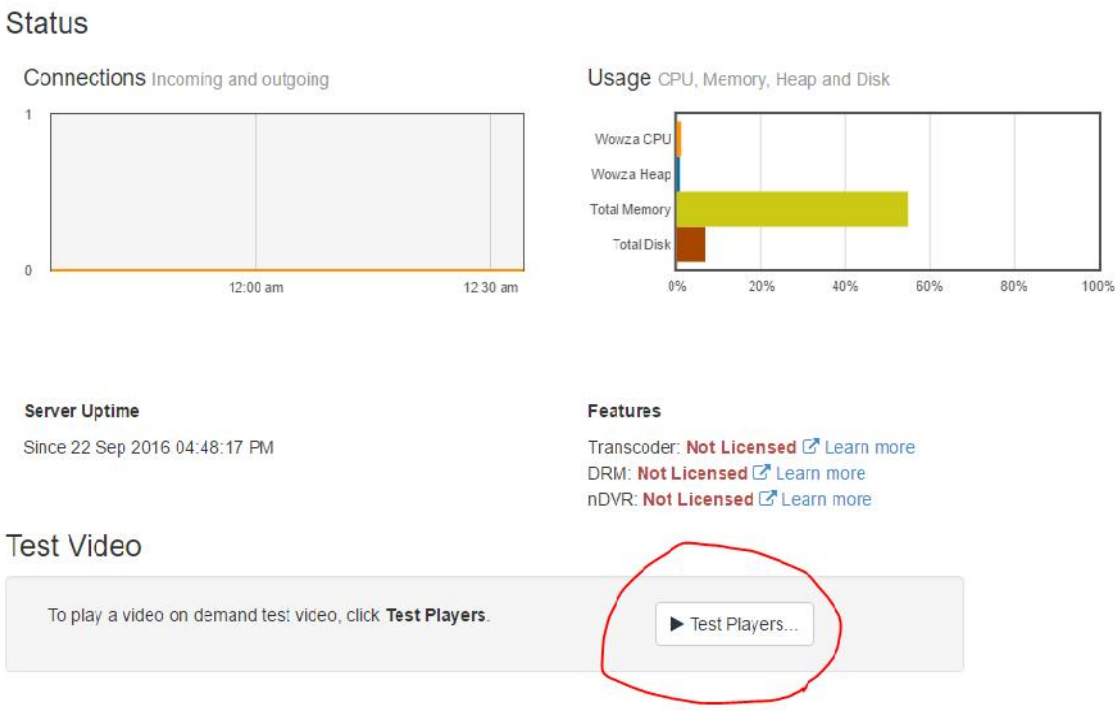
nature_960x540_2250.mp4



11. Stream the video through wowza

Thus far, you have seen how the solution stack is used to upload media files using Aspera and transcode the file using Azure Media Services. In the next section we will use the Wowza portal to the test the transcoded videos by playing them.

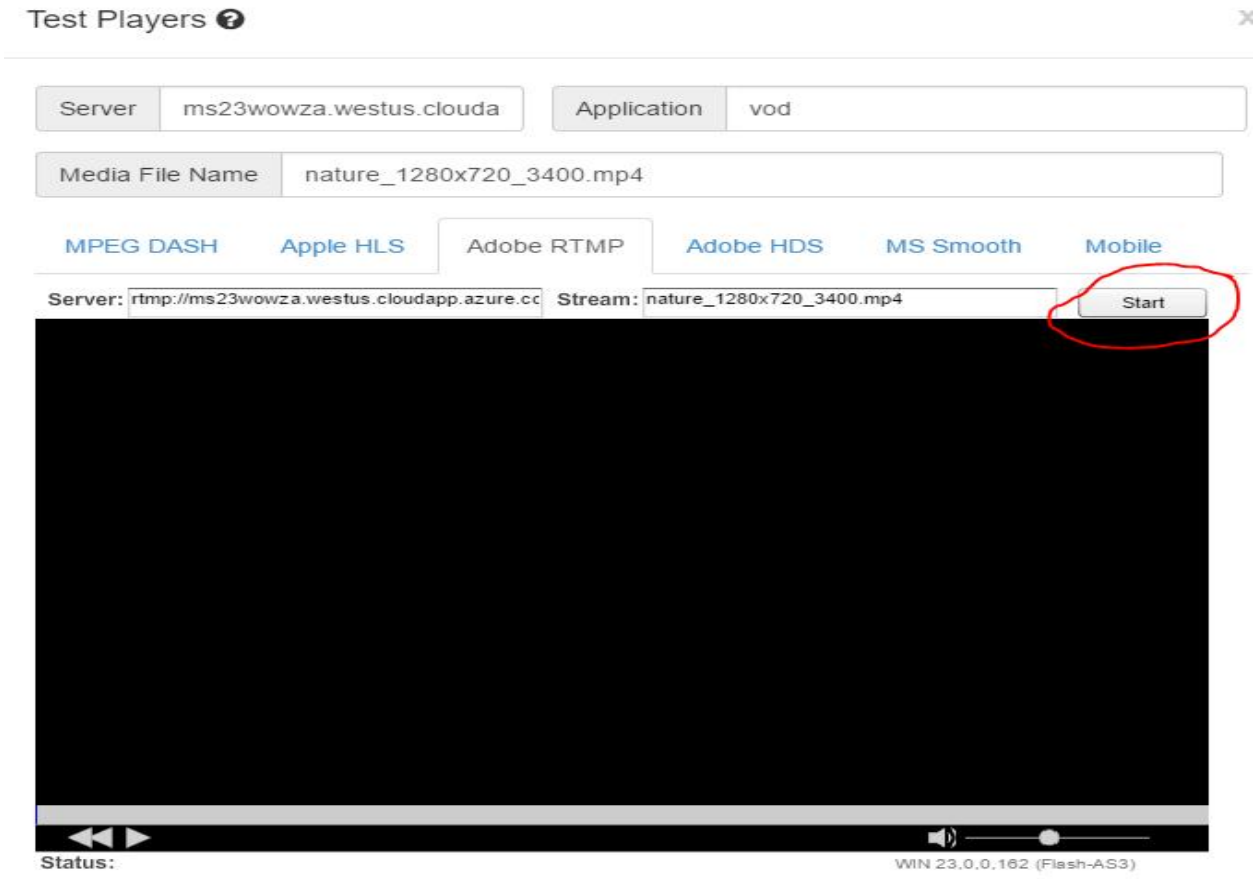
In the Wowza portal window click on “Test Players” button as shown below.

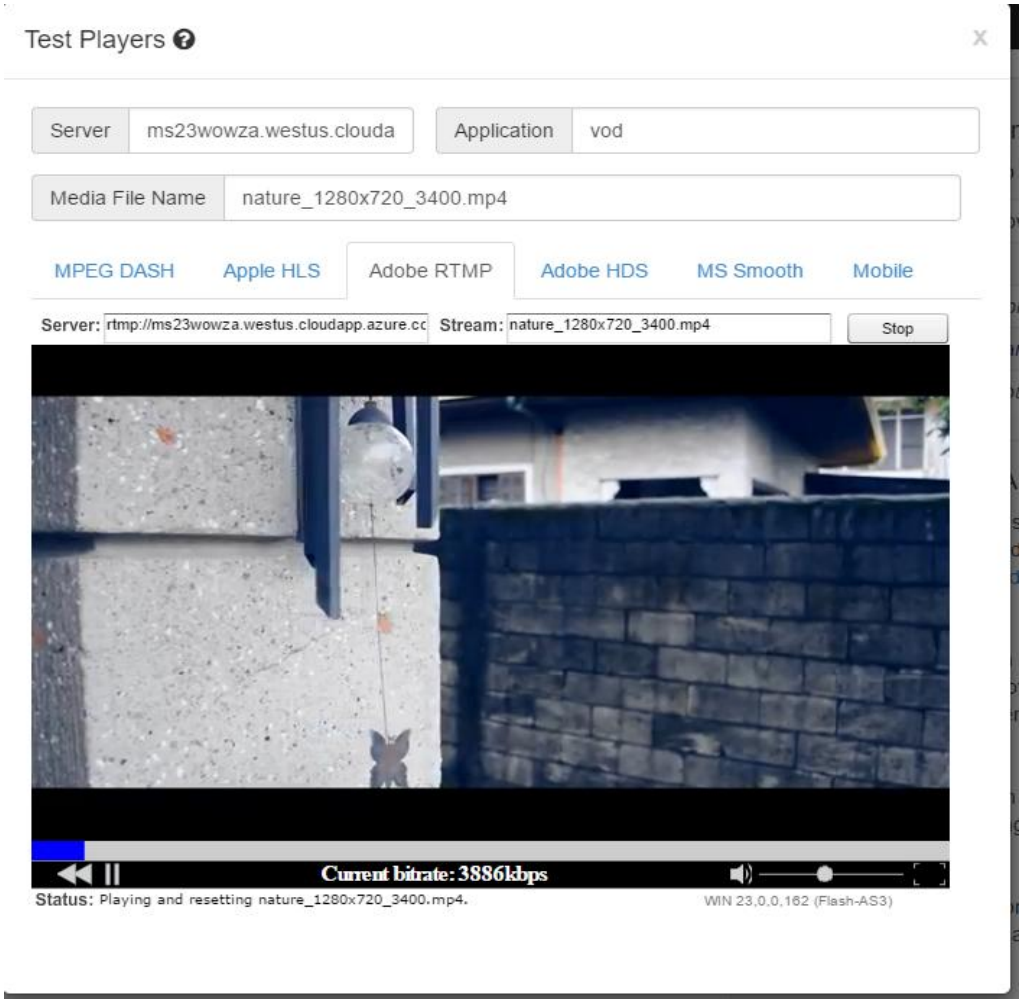


In the final step, in the server section provide wowza server information and in the media file name provide one of the file names which was transcoded from the list below and click start. The wowza streaming player would play the trasncoded file.

nature_1280x720_3400.mp4
nature_320x180_400.mp4
nature_640x360_1000.mp4

nature_640x360_650.mp4
nature_960x540_1500.mp4
nature_960x540_2250.mp4





This is the end of this hands-on lab. Hope you gained more knowledge about Azure Quickstarts and the Media Service Solution Stack. If you have questions or comments, please reach out to us at support@sysgain.com.

Thank you.