**Al Jazeera VOD Application**

Deployment Guide

Ver 3, October 30, 2013 by Brian Franklin

# Overview

The Al Jazeera VOD is an API-driven video website created in Java using the Spring framework. The content of the VOD is partially defined in server-side configuration, and party fetched dynamically from API services hosted by Brightcove, Gigya, Al Jazeera, and Taboola. The VOD is intended to be deployed on multiple application servers in a globally distributed manner. This document will define hosting requirements and suggested deployment process.

**Table of Contents**

[Overview](#h.r3c2uydyni6u)

[Infrastructure](#h.ilcj7ctaaoiu)

[Deploying a New Amazon Elastic Beanstalk Application:](#h.jmhmg1c380m8)

[Redeploying an Amazon Elastic Beanstalk Application:](#h.j6fu07rwto79)

[Deploying Directly to a Server or VM:](#h.mosvdetsfup4)

[Enabling UTF-8 on Standalone Servers](#h.4n6xz6hgsbon)

[Additional Considerations](#h.dfr05n58a1fu)

# Infrastructure

The VOD’s infrastructure only has a single server type: the application server. In lieu of hosted databases and complex infrastructure, the VOD software leverages APIs for placement and retrieval of data. The VOD is built in a way that allows horizontal scaling through the addition of application servers to the application server pool behind one or more load balancers.

While there is no strict hardware requirement, it is highly advised that each application server have 4GB+ RAM. CPU speed can vary greatly and can be mitigated through the use of additional servers. Finding the right balance between scaling up (increasing the power of the servers themselves) and scaling out (adding additional servers) is highly dependent on Al Jazeera’s global deployment and expected traffic. If deploying to Amazon EC2 or Elastic Beanstalk, a minimum server size of m1.medium is suggested. **Note that enhanced network speeds are achieved at size m1.xlarge and above, so it is highly recommended that servers of this size be used.**

All servers must have the following:

* 64 bit hardware and 64 bit OS (CentOS 6+, Amazon Linux, or RHEL 6+ advised)
* Java JDK 1.6+
* Tomcat 7+

If using Amazon AWS, the "64 bit Amazon Linux running Tomcat 7" container profile meets these requirements.

## Deploying a New Amazon Elastic Beanstalk Application:

1. If you have not previously deployed an environment for the VOD, click the "Launch New Environment" in the top right of the Elastic Beanstalk control panel.
2. You will have to specify an Environment Name and associated subdomain for your instance.
3. In the "Version" section of the window, select "Upload and use a new application version"
4. In the file selection box, select the Al-Jazeera-VOD-1.0.0.war file from your local machine
5. For the container type, select "64 bit Amazon Linux running Tomcat 7"
6. Continue through steps until your environment has been created
7. Once your environment is created, it may be a few minutes before your URL is accessible to verify your deployment.

## Redeploying an Amazon Elastic Beanstalk Application:

1. Click the "Actions" drop-down next to the environment you want to redeploy or upgrade
2. Click "Deploy a Different Version”
3. Click "Upload and deploy a new version"
4. Enter a version label (BCC suggests that this label matches the release version of the WAR file you're deploying, such as "1.0.0").
5. Select the WAR file from your local machine using the file selector.
6. Click "Deploy Version" button
7. Once your upload is complete, your environment will be deployed and the VOD web application will be restarted on the instance(s) managed by your Elastic Beanstalk. This may take a few minutes as changes propagate.

## Deploying Directly to a Server or VM:

1. Sign into the machine via SSH (Linux) or RDP (Windows).
2. Locate your Tomcat web application directory (On AWS servers, this should default to **/usr/share/tomcat7/webapps**, on Windows the default installation directory for Tomcat can be found in **Program Files**).
3. Copy the Al-Jazeera-VOD-1.0.0.war file into this **webapps** directory.
4. Rename the war file to **ROOT.war**

### Enabling UTF-8 on Standalone Servers

In order for your application servers to handle international text in UTF-8, you will need to update your Tomcat server.xml file. If you are using Elastic Beanstalk, the WAR deployment process will take care of this for you, but if deploying to a server or non-AWS virtual machine, the following steps will have to be taken:

1. Locate the **server.xml** file in **/etc/tomcat7**. In Windows Tomcat installations, you should find the server.xml file in the Tomcat installation directory.
2. Open this file in a text editor and search for "Connector". You should see a line that looks like this (note that your connector may look slightly different, this example is taken from an Elastic Beanstalk instance):

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443"/>

1. Add a URIEncoding attribute to this Connector and save your changes:

<Connector port="8080" protocol="HTTP/1.1"

connectionTimeout="20000"

redirectPort="8443"

URIEncoding="utf-8"/>

You may need to restart your Tomcat service following server.xml changes. After your Tomcat service restarts and/or your application server reboots, your VOD should correctly handle UTF-8 URLs.

To verify that your build has been updated, visit the application URL for your server or Elastic Beanstalk instance. You should see the VOD home page, featuring a version identifier in the header. The version identifier will make it easy to verify that the version seen matches the version deployed.

**Amazon AWS and EBExtensions**

Amazon Elastic Beanstalk supports the concept of “ebextensions.” These are commands that will execute on Elastic Beanstalk provisioning. In order to simplify deployment, the above server.xml configuration process has been made into an ebextension. This configuration is defined in the main/webapp directory in the VOD WAR file. The server-update.config file specifies that the server.xml replacement file should be used for Tomcat instances on the provisioned server, and the server.xml file is a modified AWS server configuration from a Tomcat deployment with the previously mentioned URIEncoding update already in place.

## Additional Considerations

**Firewalls & API Access**

The use of third-party service APIs requires the ability for the VOD to reach endpoints on the internet. This means that if the VOD is deployed behind a firewall, the following outbound domains must be whitelisted for access:

* **feeds-custom.aljazeera.net** used to serve scheduling data for EPG
* **\*.brightcove.com** used to retrieve video metadata
* **\*.gigya.com** used to set and retrieve user profile data and user curated item data

**Load Balancers**

It is assumed that one or more load balancers will be used to spread requests across multiple instances of the VOD application. **Load balancers must have “sticky” sessions enabled such that successive loads of the VOD by the same user will utilize the same server to fulfill requests.** This is extremely important to ensuring that the user’s signed-in state is persistent and that the user does not continually log out while navigating the site.

**DNS & Distribution**

Given the global nature of the VOD’s target market, it is highly suggested that the application is deployed globally. For global deployment to have the greatest effect, use of location aware DNS will allow users in different regions to be connected to the nearest hosting pool. This will decrease latency and improve performance, but configuration of location-aware DNS is beyond the scope of this document and is highly dependent on Al Jazeera’s existing infrastructure.