Artsy's Technology Stack

Site Function/Overview

Artsy is a website that is designed to find, explore, and learn more about fine art. It's mission is to "Make all of the world's art accessible to anyone with an internet connection". They have established partnerships with over 400 galleries from around the world.

Application Servers

Artsy uses Heroku, which is a Cloud Platform as a Service. This service uses a cloud and spreads it's application "dynos" across its "dyno grid", which consists of several servers. All applications run from the Heroku server use the Heroku DNS server to direct to the domain for the application, which can be something like "nameofapp.herokuapp.com". Heroku's git server handles repository pushes from permitted users.

Job/Utility Servers

All job queues are handled by Amazon's EC2, which is an "Elastic computing cloud". It's elastic because it allows the user to create, launch and end server instances as needed, so you only need to hourly for the servers that are active. EC2 uses Elastic Compute Units (ECU), each of which provide the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor. It has Elastic Block Service with volumes that can be up to 1 TB in size

Search

Search is powered by Apache Solr, which uses the Lucene Java search library for search and indexing. It has REST-like HTTP/XML and JSON APIs that make it compatible with most any programming language. The site's initial search function was powered by mongoid_fulltext, but Solr has since replaced it.

Core API

The site communicates with the "nervous system" of Artsy, which is a RESTful API built in Ruby on Rails and Grape

Front-End

The site is a responsive Backbone.js javascript app writtin in CoffeeScript/SASS and served from a Rails back-end. Jammit, an industrial strength asset packaging library for Rails, provides the CSS and Javascript concatenation and compression. It then deploys to the Amazon S3

The site relies heavily on Heroku's add-on, Memcache, which is free and open source. It's a "high performance, distributed memory object caching system… intended for use in speeding up dynamic web applications by alleviating database load.

For email, SendGrid and MailChimp are used. New Relic and Pingdom are used in combination to monitor the systems, and Jenkins was used to build, test and deploy everything.