

# Avalon on AWS

Phase 1: simple Avalon deployment, each component on one EC2 instance.

## Amazon VPC

Create 1 VPC with 1 public and 2 private subnets (in 2 different availability zones) (Amazon RDS requires 2+ availability zones)

- 1 security group for public subnet, allows 8080, 8983 and SSH.
- 1 security group for private subnet, allows HTTP(S) and SSH.

## Amazon RDS

Create new DB subnet group from previous 2 private subnets

Create new RDS instance with MySQL engine

## Amazon EC2

1 small instance for Avalon web server

4 micro instances for backend components: Matterhorn, Fedora, Red5, Solr

## Matterhorn

Regular install [Manual Installation Instructions#Matterhorn](#)

## Fedora

Install Tomcat 7.0.70 [Setup a Server for Hydra 10 Development#InstallFedora4.5.1](#)

Create Fedora db on Amazon RDS instance

Create fcrepo user and grant access to Fedora db

Install Fedora on existing Tomcat [Fedora \(standalone\)](#) (change JDBC url to Amazon RDS MySQL)

## Solr

Deploy Solr Docker on Elastic Beanstalk

## Amazon Elastic Transcoder

Matterhorn and Red5 can be replaced by AET & CloudFront. Steps:

- Create AET pipeline and presets
- Create CloudFront HLS & RTMP streaming directories
- Create ActiveEncode engine adapter for AET.
- Rewrite ActiveEncodeJob that uploads file from Rails server to S3 and polls for progress.
- Modify active\_encode initializer and MasterFile model to use AET.
- Modify Avalon config to use CloudFront streaming
- Figure out [stream auth](#)

Notes:

- Dropbox: can use Amazon Lambda function to monitor and ingest

- Can use Amazon SNS and SQS for notifications and polling.
- May upload directly to S3 (not through Rails server)