

Web (standalone)



This documentation has not been updated for Release 4.0.0 yet. Please see the one host (all-in-one) [Manual Installation Instructions](#).



This documentation is for Release 3.2. For the Release 1 version of this page, see [v.81](#). For the Release 2 version of this page, see [v.87](#). For the Release 3.0-3.1 version of this page, see [v.15](#).

Intended as guidance for production environments with different hosts for most tasks, these instructions provide a recipe for building only the **Web** piece of an Avalon system from scratch on CentOS or Red Hat Enterprise Linux. Recipes for the other components for Avalon may be found here. [Fedora \(standalone\)](#), [MySQL \(standalone\)](#), [Solr \(standalone\)](#), [Red5 \(standalone\)](#), and [Matterhorn \(standalone\)](#).

Instructions for building every component on one host (all-in-on) please see here: [Manual Installation Instructions](#).

- [Ready the Installation Environment](#)
- [HTTPD](#)
- [Apache Passenger and Ruby](#)
- [Avalon](#)
- [Using the System](#)
- [Restarting the Server](#)

Ready the Installation Environment

Install EPEL

```
rpm -ivh http://linux.mirrors.es.net/fedora-epel/6/i386/epel-release-6-8.noarch.rpm
```

This package has libyaml-devel which is required by ruby and not provided by Redhat.

Make sure a valid hostname is resolvable

The default hostname is "avalon.dev", so name the machine this and enter it into /etc/hosts

```
# hostname
avalon.dev
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4 avalon.dev
```

Configure iptables

The Avalon Media System requires several ports to be open to client browsers.

Here are the port settings that will need to be configured:

Port	Purpose	External?
80	HTTP (Avalon)	Yes

The preferred method is to create a shell script that will do the work for you. Here is an example script that you should look through and customize as needed: [avalon-iptables-config.sh](#)



If you're connected over ssh, it might kick you off.

Save your script to /etc/sysconfig/avalon-iptables-config.sh, make it executable and run it.

```
chmod +x /etc/sysconfig/avalon-iptables-config.sh
/etc/sysconfig/avalon-iptables-config.sh
```

If you run into connection issues you can disable the iptables, by running "service iptables stop". This will completely drop your firewall. When finished troubleshooting run "service iptables start".

Disable SELinux

```
echo 0 > /selinux/enforce
vim /etc/selinux/config #change the value of `SELINUX` from `enforcing` to `permissive`
```

Add the NUL repository

Create the NUL repository config file:

```
vim /etc/yum.repos.d/nul-public.repo
```

Append the following code:

```
[nul_public]
name=NUL Library Public RHEL repository
baseurl=http://yumrepo-public.library.northwestern.edu/x86_64
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-nul
cost=150
```

Install and place the NUL GPG key in the proper location:

```
curl http://yumrepo-public.library.northwestern.edu/RPM-GPG-KEY-nul -o /etc/pki/rpm-gpg/RPM-GPG-KEY-nul
```

Install development libraries and packages for building Ruby

```
yum groupinstall "Development Tools"
yum install readline-devel zlib-devel libyaml-devel libffi-devel openssl-devel libxml2-devel libxslt-devel
```

HTTPD

Install and start the httpd service.

```
yum install httpd
service httpd start
```

Add avalon user and create avalon directory.

```
useradd avalon
mkdir /var/www/avalon
chown -R avalon:avalon /var/www/avalon
```

Apache Passenger and Ruby

Change current user to avalon then install RVM and ruby 2.1.5.

```
su - avalon
curl -L https://get.rvm.io | bash -s stable --ruby=2.1.5
```

Source the RVM shell (as avalon user)

```
source /home/avalon/.rvm/scripts/rvm
```

Install Passenger via Gem (as avalon user)

```
gem install passenger
```

Check to make sure passenger installed in the expected location (as avalon user)

```
$ passenger-config --root
/home/avalon/.rvm/gems/ruby-2.1.5/gems/passenger-4.0.56
```

Install Passenger apache module requirements (as root)

```
su - root
yum install curl-devel httpd httpd-devel apr-devel apr-util-devel
```

Build passenger for your version of Apache and Ruby (as avalon user)

```
su - avalon
passenger-install-apache2-module #copy the suggested Apache configuration file settings for later
```

Create an apache configuration file (as root)

```
su - root
vim /etc/httpd/conf.d/passenger.conf
```

Example contents of /etc/httpd/conf.d/passenger.conf:

```
LoadModule passenger_module /home/avalon/.rvm/gems/ruby-2.1.5/gems/passenger-4.0.56/buildout/apache2
/mod_passenger.so
<IfModule passenger_module>
  PassengerRoot /home/avalon/.rvm/gems/ruby-2.1.5/gems/passenger-4.0.56
  PassengerDefaultRuby /home/avalon/.rvm/wrappers/ruby-2.1.5/ruby
  PassengerMaxPoolSize 30
  PassengerPoolIdleTime 300
  PassengerMaxInstancesPerApp 0
  PassengerMinInstances 3
  PassengerSpawnMethod smart-lv2
</IfModule>
```

Apache security configuration

```
wget https://raw.githubusercontent.com/avalonmediasystem/config-files/master/sbin/avalon_auth -O /usr/local/sbin
/avalon_auth
chmod +x /usr/local/sbin/avalon_auth
wget https://raw.githubusercontent.com/avalonmediasystem/config-files/master/apache/10-mod_rewrite.conf -P /etc/httpd/conf.
d/
```

Create a virtual host for avalon in /etc/httpd/conf.d/avalon.conf

```
wget https://raw.githubusercontent.com/avalonmediasystem/config-files/master/apache/20-avalon.conf -P /etc/httpd/conf.d/  
vim /etc/httpd/conf.d/20-avalon.conf
```

Add this line to 20-avalon under the VirtualHost tag:

```
RailsEnv development
```

Restart apache. With apache running, check passenger-status

```
/etc/init.d/httpd restart  
su - avalon  
which passenger-status  
#> ~/.rvm/gems/ruby-2.1.5/bin/passenger-status
```

Avalon

Grab Avalon code from github

```
cd ~  
git clone git://github.com/avalonmediasystem/avalon.git  
cd avalon  
git checkout master #make sure you are in the master branch (should be by default)  
mv public/* /var/www/avalon/public/  
mv * /var/www/avalon/
```

Configure Avalon

Create `/var/www/avalon/config/setup_load_paths.rb` and add:

```
if ENV['MY_RUBY_HOME'] && ENV['MY_RUBY_HOME'].include?('rvm')  
  begin  
    gems_path = ENV['MY_RUBY_HOME'].split(/@/)[0].sub(/rubies/, 'gems')  
    ENV['GEM_PATH'] = "#{gems_path}:#{gems_path}@global"  
    require 'rvm'  
    RVM.use_from_path! File.dirname(File.dirname(__FILE__))  
  rescue LoadError  
    raise "RVM gem is currently unavailable."  
  end  
end  
# If you're not using Bundler at all, remove lines bellow  
ENV['BUNDLE_GEMFILE'] = File.expand_path('../Gemfile', File.dirname(__FILE__))  
require 'bundler/setup'
```

```
cd /var/www/avalon/config  
cp authentication.yml.example authentication.yml
```

Configure database settings

```
vim database.yml
```

Replace database.yml with the correct values for your development environment

```
development:
  adapter: mysql2
  host: localhost
  database: rails
  username: rails
  password: rails
  pool: 5
  timeout: 5000
```

Install the mysql2 adapter

```
su - root
yum install mysql-devel
su - avalon
gem install activerecord-mysql2-adapter
gem install mysql2
su - root
vim /var/www/avalon/Gemfile
```

Add this line to the Gemfile

```
gem 'mysql2', '~>0.3.11'
```

If you are using mysql instead of sqlite (as is done throughout this document), then comment out the following lines in the Gemfile

```
# gem 'activerecord-jdbcsqlite3-adapter'
# gem 'jdbc-sqlite3'
# gem 'sqlite3'
```

Run the bundle install

Install additional dependencies

```
yum install rubygem-json cmake
```

Run bundler

```
su - avalon
cd /var/www/avalon
gem update debugger-ruby_core_source
bundle install
```

Finish configuring Avalon

Edit */var/www/avalon/config/solr.yml*

```
development:
  url: http://localhost:8983/solr/
```

Edit */var/www/avalon/config/fedora.yml*

```
development:
  user: fedoraAdmin
  password: fedoraPassword
  url: http://127.0.0.1:8983/fedora
```

Create */var/www/avalon/config/matterhorn.yml*

```
development:
  url: http://matterhorn_system_account:CHANGE_ME@localhost:8080/
```

Create `/var/www/avalon/config/avalon.yml` and base it off of `/var/www/avalon/config/avalon.yml.example`. Consult the [documentation](#) to customize this file for your installation.

```
development:
  dropbox:
    path: '/var/avalon/dropbox/'
    upload_uri: 'sftp://localhost/var/avalon/dropbox'
    username: 'test'
    password: 'test'
    notification_email_address: ''
```

Create the database using rake

```
su - avalon
cd /var/www/avalon
rake db:create
```

If you get an error message saying that you can't connect to the database, take a look at this post and follow some of the troubleshooting steps.

<http://stackoverflow.com/questions/5376427/cant-connect-to-local-mysql-server-through-socket-var-mysql-mysql-sock-38>

Run the database migrations

```
rake db:migrate
```

Set rails environment to development, if it has not defaulted to this. On the first line of `/var/www/avalon/config/environment.rb` make sure it says 'development'

```
ENV['RAILS_ENV'] ||= 'development'
```

Visit your new Avalon site!

You should be able to visit the webpage with just the hostname (ie <http://localhost>)

Click on "Sign in" in the upper right corner of the website main page. Set up a default identity with the following properties.

```
archivist1@example.com
password/password
```

This is a known identity with administrative privileges.

Start delayed_job

As avalon run

```
cd /var/www/avalon/
bundle exec rake delayed_job:start
```

Delayed Job handles background jobs. DelayedJob logs to `log/delayed_job.log` in the avalon directory.

Using the System

You can find specific information about using the system in the [Collection Manager's Guide](#). Upload items individually or via batch. [Sample content](#) is available for your convenience.

[Known Issues](#) - a list of bugs, workarounds, and cautions.

Restarting the Server

Before you restart your Avalon server, you'll want to make sure all of the services necessary to run Avalon will start automatically after the restart. Run these commands once and you should be set:

```
chkconfig --level 345 sshd on  
chkconfig --level 345 httpd on
```