

Upgrading Avalon 3.1 to Avalon 3.2

Preparing for the Upgrade

Before doing anything to prepare for the upgrade, backup your Avalon 3.1 installation.

See [What's New in Avalon 3.2](#) for more details.

Instructions

The following instructions will work for any Avalon 3.1 installation that used Puppet as its source. This includes the [portable VM Image Installation](#), the [Vagrant Virtual Machine Installation](#), and the [Manual Installation of Avalon Using Puppet](#).

Upgrading is fairly simple, and must be run as the root user from a bash prompt on the Avalon system. The process for this depends on which kind of installation you have.

- **Portable VM image:**
Open a terminal window from within the VM console, and type `su root` to become the root user. This will require the root password you provided the first time the system was booted.
- **Vagrant virtual machine:**
From within the original installer directory (from step 4 of the [Vagrant Virtual Machine Image Installation Instructions](#)), type `vagrant ssh` to log in as the vagrant user, then type `sudo -i` to become the root user.
- **Manual Puppet installed machine:**
Log in and become root as you did in steps 1 and 2 of the [Manual Installation of Avalon Using Puppet](#) instructions.

Once you have a root prompt, proceed with the upgrade as follows:

Upgrade Avalon

```
# 1. if starting from Portable VM image, download the installer package, unpack, and link it
$ wget http://www.avalonmediasystem.org/downloads/avalon-installer-flat.tar.gz -O flat.tar.gz
$ tar xzf flat.tar.gz
$ export INSTALL_DIR=`pwd`
$ ln -s $INSTALL_DIR/avalon-installer-flat/files /etc/puppet/avalon_files

# 2. navigate to the installer
$ cd avalon-installer-flat

# 3. remove the Avalon deploy tag
$ rm -f /var/www/avalon/bare-deploy
$ rm -rf /opt/staging/avalon/avalon-bare-deploy.tar.gz /opt/staging/avalon/avalon-bare-deploy

# 4. apply the puppet manifest
$ puppet apply --fileserversconfig=fileserver.conf --modulepath=modules \
  --hiera_config=hiera/hiera.yaml --templatedir=templates manifests/init.pp

# 5. add configuration for new controlled vocabulary
# (Merge new identifier type definitions from config/controlled_vocabulary.yml.example into config
# /controlled_vocabulary.yml and edit as necessary.)

# 6. reindex
$ su - avalon -c "cd /var/www/avalon/current; RAILS_ENV='production' bundle exec rake avalon:reindex"
```

Upgrade Matterhorn

```
# 1. Switch to matterhorn user
$ su - matterhorn
$ cd /usr/local/matterhorn

# 2. Install updated modules
$ cd lib/matterhorn
$ wget --no-check-certificate https://raw.githubusercontent.com/avalonmediasystem/avalon-felix/1.4.x/lib/matterhorn/matterhorn-distribution-service-hls-1.4-SNAPSHOT.jar -O matterhorn-distribution-service-hls-1.4-SNAPSHOT.jar

# 3. Install updated encoding profiles
$ cd ../../etc/encoding
$ wget --no-check-certificate https://raw.githubusercontent.com/avalonmediasystem/avalon-felix/1.4.x/etc/encoding/avalon.properties -O avalon.properties

# 4. Restart Matterhorn
$ exit
$ service matterhorn stop
$ service matterhorn start
```

If you didn't load the sample content before, [try it out](#) to test your avalon 3.2 upgrade.