Upgrading Avalon 3.0 to Avalon 3.1

Preparing for the Upgrade

There are some steps you can take to make the upgrade and transition to Avalon 3.1 smoother. Before doing anything to prepare for the upgrade, backup your Avalon 3.0 installation.

LDAP groups can now be used to grant access to items. See LDAP Access Control for how to set this up.

See What's New in Avalon 3.1 and the release notes for more details.

Instructions

The following instructions will work for any Avalon 3.0 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 R3 installer
$ cd avalon-installer-flat

# 3. remove the Avalon deploy tag
$ rm -f /var/www/avalon/bare-deploy

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

# 5. reindex
$ su - avalon -c "cd /var/www/avalon/current; RAILS_ENV='production' 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-workflow-service-api-1.4-SNAPSHOT.jar -O matterhorn-workflow-service-api-1.4-SNAPSHOT.jar
$ wget --no-check-certificate https://raw.githubusercontent.com/avalonmediasystem/avalon-felix/1.4.x/lib
/matterhorn/matterhorn-workflow-notifier-avalon-1.4-SNAPSHOT.jar -O matterhorn-workflow-notifier-avalon-1.4-
SNAPSHOT.jar

# 3. Install updated workflows
$ cd ../../etc/workflows
$ wget --no-check-certificate https://raw.githubusercontent.com/avalonmediasystem/avalon-felix/1.4.x/etc
/workflows/avalon-video.xml -O avalon-video.xml
$ wget --no-check-certificate https://raw.githubusercontent.com/avalonmediasystem/avalon-felix/1.4.x/etc
/workflows/avalon-skip-transcoding.xml -O avalon-skip-transcoding.xml

# 4. Update log configuration
$ echo "log4j.logger.org.avalonmediasystem=INFO" >> /usr/local/matterhorn/etc/services/org.ops4j.pax.logging.
properties

# 5. 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 3.1 upgrade.