

A MuseWiki

Upgrading

Marco Pessotto

Marco Pessotto
Upgrading

amusewiki.org

Contents

Debian package	3
Running in tree with the init script	3
Running in tree, but started with systemd and selinux	4

The upgrade procedure depends on the way you installed amusewiki.

Debian package

The recommended way is to install the application from the debian packages. That way the upgrade is automatically managed by the package, with zero downtime.

Please note that if you are using mysql or postgresql, and the database is currently being restarted by the upgrade procedure, you will get a failure during the amusewiki upgrade. This is nothing to be scared. Just run `apt-get upgrade` again and it will be fixed (again, without the application going down).

Running in tree with the init script

If you installed from git and are running the app in the tree:

```
./init-all.sh stop
git pull
./init-all.sh start
```

DB upgrades are managed automatically. If some manual operation is required, it will be listed in the Changes file.

This is the suggested script to do a 0-downtime upgrade. We don't ship it in the distribution because it looks fragile to run a shell script which potentially could change itself.

```
#!/bin/sh
```

```
set -e
```

```
./script/amusewikifarm_fastcgi.pl -l var/amw.sock -n 3 2>/dev/null >
temp_pid=$!
echo "Started copy of the application with pid $temp_pid. If this sc
echo "Waiting for the backup app to come up"
sleep 20
echo "Stopping the app"
./init-all.sh stop
```

```

git pull
echo "Starting the app"
./init-all.sh start
echo "Waiting for the new app to come up"
sleep 20
# kill the old
echo "Stopping the backup app $temp_pid"
kill $temp_pid
echo "All done, no errors"

```

Running in tree, but started with systemd and selinux

You could install this script and use it to perform the upgrades (or do manually with the same logic). Run it as root.

```

#!/bin/sh

set -e

amusewiki=/var/www/amusewiki/amusewiki
if [ "x$1" != "x" ]; then
    amusewiki=$1
fi

if [ -f "$amusewiki/lib/AmuseWikiFarm.pm" ]; then
    owner=`ls -ld "$amusewiki/lib/AmuseWikiFarm.pm" | awk '{print $3}'`
else
    echo "$amusewiki is not a directory"
    exit 2
fi

if [ "$owner" = "root" ]; then
    echo "Owner of amusewiki shouldn't be root";
    exit 2;
fi

if [ "x$owner" = "x" ]; then

```

```

    echo "Amusewiki owner not found!"
    exit 2
fi

if id $owner > /dev/null; then
    echo "Upgrading $amusewiki owned by $owner"
    sleep 2
else
    echo "$owner not found"
    exit 2
fi

systemctl stop amusewiki-jobber
su -c "cd $amusewiki; git pull; ./init-all.sh start-app" -l $owner
sleep 10;
systemctl restart amusewiki-web
sleep 10;
su -c "cd $amusewiki; ./init-all.sh stop-app" -l $owner
sleep 10;
systemctl start amusewiki-jobber
systemctl restart amusewiki-cgit
sleep 10;
# check if everything is fine
systemctl status 'amusewiki-*'
ps -u $owner f

```