Backing up this Wiki

The backup method described at Automated Backup requres settings to be done on the source server. This wiki, however, is backed up by method where the backup is activated from the backup server and no settings are needed on the source server. This is particularly useful when backing up multiple servers from a single backup server. Here is the script, which you can change to suit your particular purpose.

```
#!/bin/sh
#=======================
# Placed in the /root folder of the backup server
# backup script for stuff on wiki to be run periodically through a crontab
on a trusted server
# > crontab -e
#
# daily : 1 1 * * * /path/to/script/wwwbackup.sh
# weekly: 1 1 * * 0 /path/to/script/wwwbackup.sh # I Guess weekly backup is
enough for not frequetly updated wiki
#=======================
# here=`pwd`
pathname=$(readlink -f "$0")

mydate="`date '+%Y%m%d.%H%M'`"

# Define wiki location
myhost='wiki.dreamapps.com'                        #name of the server
hosting your your wiki
myuser='root'                       #name of the user that has an ssh access
to "myhost"
relpathtowiki='/srv/www/htdocs/wiki/'   #relative path to your wiki from
"myhost" base
backupsource="${myuser}@${myhost}:${relpathtowiki}"
#backupsource="/abs/path/to/wiki/" #Use this line instead of the above if
you run the backup script directly on your www server.

# Define location of backup
backupdir="/home/ds1184/bu/${myhost}/wiki"
logfile="${backupdir}/backup.log"
excludelist="${pathname}-exclude.txt"
bkname="backup"

nbbackup="7" # keep this amount of old backup

#-- creates $1, if not existant
checkDir() {
    if [ ! -d "${backupdir}/$1" ] ; then
        mkdir -p "${backupdir}/$1"
    fi
```

```
}

# 1 -> path
# 2 -> name
# 3 -> number of backups
rotateDir() {
    for i in `seq $(($3 - 1)) -1 0`
      do
      if [ -d "$1/$2-$i" ] ; then
          /bin/rm -f "$1/$2-$((i + 1))"
          mv "$1/$2-$i" "$1/$2-$((i + 1))"
      fi
    done
}

#-- make sure everything exists
checkDir "archive"
checkDir "daily"

#-- first step: rotate daily.
rotateDir "${backupdir}/daily" "$bkname" "$nbbackup"

mv ${logfile} ${backupdir}/daily/${bkname}-1/

cat >> ${logfile} <<_EOF
==========================================
  Backup done on: $mydate
==========================================
_EOF

#-- Do the backup and save difference in backup-1
mkdir -p ${backupdir}/daily/${bkname}-1/
mkdir -p ${backupdir}/daily/${bkname}-0/
cd ${backupdir}/daily/${bkname}-0
rsync -av --whole-file --delete --force \
    -b --backup-dir ${backupdir}/daily/${bkname}-1/${thisdir} \
    --exclude-from=${excludelist} \
    $backupsource . \
    1>> ${logfile} 2>&1

#-- create an archive backup every month
lastarchivetime="0"
if [ -r ${backupdir}/lastarchivetime ] ; then
  lastarchivetime=`cat ${backupdir}/lastarchivetime`
fi
now=`date +%j`
let diffday=$now-$lastarchivetime
if [ $diffday  -ge 30 -o $diffday -lt 0 ] ; then
    echo $now > ${backupdir}/lastarchivetime
    cd ${backupdir}/daily
    tar -cjf ${backupdir}/archive/${bkname}-${mydate}.tar.bz2 ./${bkname}-0
```

```
fi
```

Generating and loading the key

In the back up server

```
ssh-keygen -t rsa
ssh root@destination.dreamapps.com  mkdir -p .ssh
cat .ssh/id_rsa.pub | ssh root@destination.dreamapps.com 'cat >>
.ssh/authorized_keys'
```