Minecraft Server Launcher v1.0

参考mc-wiki-launcher

```bash
#!/bin/bash
# /etc/init.d/minecraft
# version 0.4.1 2015-05-07 (YYYY-MM-DD)
#
### BEGIN INIT INFO
# Provides:   minecraft                               #提供者：Minecraft
# Required-Start: $local_fs $remote_fs screen-cleanup#需要的开始
# Required-Stop:  $local_fs $remote_fs                #需要的结束
# Should-Start:   $network                            #应该的开始
# Should-Stop:    $network                            #应该的结束
# Default-Start:  2 3 4 5                             #默认启动
# Default-Stop:   0 1 6                               #默认结束
# Short-Description:    Minecraft server              #简介
# Description:    Starts the minecraft server         #
### END INIT INFO

#Settings
SERVICE='minecraft_server.jar'                     #服务端核心名字
SCREENNAME='minecraft_server'                      #显示名
OPTIONS='nogui'                                    #jre选项
USERNAME='minecraft'                               #用户名
WORLD='world'                                      #地图名字
MCPATH='/home/minecraft'                           #mc路径
BACKUPPATH='/media/remote.share/minecraft.backup'  #备份路径
MAXHEAP=2048                                       #最大内存
MINHEAP=1024                                       #最小内存
HISTORY=1024                                       #历史
CPU_COUNT=1                                        #CPU核数
INVOCATION="java -Xmx${MAXHEAP}M -Xms${MINHEAP}M -XX:+UseConcMarkSweepGC \
-XX:+CMSIncrementalPacing -XX:ParallelGCThreads=$CPU_COUNT -XX:+AggressiveOpts \
-jar $SERVICE $OPTIONS"                            #调用

ME=`whoami`                                        #获取当前用户（需要吗？）
as_user() {
  if [ "$ME" = "$USERNAME" ] ; then
    bash -c "$1"
  else
    su - "$USERNAME" -c "$1"
  fi
}

mc_start() {
  if  pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    echo "$SERVICE is already running!"
  else
    echo "Starting $SERVICE..."
    cd $MCPATH
    as_user "cd $MCPATH && screen -h $HISTORY -dmS ${SCREENNAME} $INVOCATION"
    sleep 7
    if pgrep -u $USERNAME -f $SERVICE > /dev/null
    then
      echo "$SERVICE is now running."
    else
      echo "Error! Could not start $SERVICE!"
    fi
  fi
}

mc_saveoff() {
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    echo "$SERVICE is running... suspending saves"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"say SERVER BACKUP STARTING. Server going readonly...\"\015'"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"save-off\"\015'"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"save-all\"\015'"
    sync
    sleep 10
  else
    echo "$SERVICE is not running. Not suspending saves."
  fi
}

mc_saveon() {
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    echo "$SERVICE is running... re-enabling saves"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"save-on\"\015'"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"say SERVER BACKUP ENDED. Server going read-write...\"\015'"
  else
    echo "$SERVICE is not running. Not resuming saves."
  fi
}

mc_stop() {
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    echo "Stopping $SERVICE"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"say SERVER SHUTTING DOWN IN 10 SECONDS. Saving map...\"\015'"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"save-all\"\015'"
    sleep 10
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"stop\"\015'"
    sleep 7
  else
    echo "$SERVICE was not running."
  fi
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    echo "Error! $SERVICE could not be stopped."
  else
    echo "$SERVICE is stopped."
  fi
}

mc_update() {
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
   echo "$SERVICE is running! Will not start update."
  else
    as_user "cd $MCPATH && wget -q -O $MCPATH/versions http://s3.amazonaws.com/Minecraft.Download/versions/versions.json"
       snap=`awk -v linenum=3 'NR == linenum {print; exit}' "$MCPATH/versions"`
       snapVersion=`echo $snap | awk -F'\"' '{print $4}'`
       re=`awk -v linenum=4 'NR == linenum {print; exit}' "$MCPATH/versions"`
       reVersion=`echo $re | awk -F'\"' '{print $4}'`
       as_user "rm $MCPATH/versions"
       if [ "$1" == "snapshot" ]; then
       MC_SERVER_URL=http://s3.amazonaws.com/Minecraft.Download/versions/$snapVersion/minecraft_server.$snapVersion.jar
       else
       MC_SERVER_URL=http://s3.amazonaws.com/Minecraft.Download/versions/$reVersion/minecraft_server.$reVersion.jar
       fi
    as_user "cd $MCPATH && wget -q -O $MCPATH/minecraft_server.jar.update $MC_SERVER_URL"
    if [ -f $MCPATH/minecraft_server.jar.update ]
    then
      if `diff $MCPATH/$SERVICE $MCPATH/minecraft_server.jar.update >/dev/null`
      then
        echo "You are already running the latest version of $SERVICE."
      else
        as_user "mv $MCPATH/minecraft_server.jar.update $MCPATH/$SERVICE"
        echo "Minecraft successfully updated."
      fi
    else
      echo "Minecraft update could not be downloaded."
    fi
  fi
}
```

```

mc_backup() {
   mc_saveoff

    NOW=`date "+%Y-%m-%d_%Hh%M"`
   BACKUP_FILE="$BACKUPPATH/${WORLD}_${NOW}.tar"
   echo "Backing up minecraft world..."
   #as_user "cd $MCPATH && cp -r $WORLD $BACKUPPATH/${WORLD}_`date "+%Y.%m.%d_%H.%M"`"
   as_user "tar -C \"$MCPATH\" -cf \"$BACKUP_FILE\" $WORLD"

   echo "Backing up $SERVICE"
   as_user "tar -C \"$MCPATH\" -rf \"$BACKUP_FILE\" $SERVICE"
   #as_user "cp \"$MCPATH/$SERVICE\" \"$BACKUPPATH/minecraft_server_${NOW}.jar\""

   mc_saveon

   echo "Compressing backup..."
   as_user "gzip -f \"$BACKUP_FILE\""
   echo "Done."
}

mc_command() {
  command="$1";
  if pgrep -u $USERNAME -f $SERVICE > /dev/null
  then
    pre_log_len=`wc -l "$MCPATH/logs/latest.log" | awk '{print $1}'`
    echo "$SERVICE is running... executing command"
    as_user "screen -p 0 -S ${SCREENNAME} -X eval 'stuff \"$command\"\015'"
    sleep .1 # assumes that the command will run and print to the log file in less than .1 seconds
    # print output
    tail -n $[`wc -l "$MCPATH/logs/latest.log" | awk '{print $1}'`-$pre_log_len] "$MCPATH/logs/latest.log"
  fi
}

#Start-Stop here
case "$1" in
  start)
    mc_start
    ;;
  stop)
    mc_stop
    ;;
  restart)
    mc_stop
    mc_start
    ;;
  update)
    mc_stop
    mc_backup
    mc_update $2
    mc_start
    ;;
  backup)
    mc_backup
    ;;
  status)
    if pgrep -u $USERNAME -f $SERVICE > /dev/null
    then
      echo "$SERVICE is running."
    else
      echo "$SERVICE is not running."
    fi
    ;;
  command)
    if [ $# -gt 1 ]; then
      shift
      mc_command "$*"
    else
```

```
      echo "Must specify server command (try 'help'?)"
    fi
    ;;

  *)
  echo "Usage: $0 {start|stop|update|backup|status|restart|command \"server command\"}"
  exit 1
  ;;
esac

exit 0
```