

service script 3

service scripts 5

Last update: 2015/12/07 03:36 https://atl.kr/dokuwiki/doku.php/%EB%A6%AC%EC%86%8C%EC%8A%A4_%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8_%EC%9E%91%EC%84%B1%EB%B2%95

URL : <https://access.redhat.com/solutions/5912>

service script

3가

- 1. service start 가 (0)
0
- 2. service stop 가 (0)
0
- 3. service status 가 (0)
0

script

```
#!/bin/bash

#
# Copyright Red Hat Inc., 2002
# Copyright Mission Critical Linux, 2000
#
# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2, or (at your option) any
# later version.
#
# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING. If not, write to the
# Free Software Foundation, Inc., 675 Mass Ave, Cambridge,
# MA 02139, USA.
#
#
# Author: Gregory P. Myrdal
#
#
# This file contains a template that you can use to create a script
# that will start, stop and monitor an application used in a cluster \
# service.
#
#-----
# Variable definitions
#-----
MYNAME=$(basename $0)
```

```
# The clug utility uses the normal logging levels as defined in
# sys/syslog.h. Calls to clulog will use the logging level defined
# for the Service Manager (clusvcmgrd).
LOG_EMERG=0 # system is unusable
LOG_ALERT=1 # action must be taken immediately
LOG_CRIT=2 # critical conditions
LOG_ERR=3 # error conditions
LOG_WARNING=4 # warning conditions
LOG_NOTICE=5 # normal but significant condition
LOG_INFO=6 # informational
LOG_DEBUG=7 # debug-level messages

#-----
# Start of execution
#-----
if [ $# -ne 1 ]; then
    echo "Usage: $0 {start, stop, status}"
    exit 1
fi

action=$1 # type of action, i.e. 'start', 'stop' or 'status'

# Record all output into a temp file in case of error
exec > /tmp/$MYNAME.$action.log 2>1

clulog -s $LOG_DEBUG "In $0 with action=$action,"

case $action in
    'start')
        clulog -s $LOG_INFO "Running user start script for service $0"
        # <<< Add service start specific code here >>>
        /path/to/my-service-start
        exit $?
        ;;
    'stop')
        clulog -s $LOG_INFO "Running user stop script for service $0"
        # <<< Add service stop specific code here >>>
        /path/to/my-service-stop
        exit $?
        ;;
    'status')
        clulog -s $LOG_INFO "Running user status script for service $0"
        # <<< Add service status specific code here >>>
        pidof my-service-process
        exit $?
        ;;
    *)
```

```

        clulog -s $LOG_ERR "Unknown action '$action' passed to $0 user
script"
        exit 1 # return failure
esac

```

service scripts

A normal start-status-stop cycle: 가 start-status-stop

```

# Service not yet start, status return failure //     가
        status     failure     . ( 0     )
$ service script_name status; echo $?
1
# Start the service, return success
$ service script_name start; echo $?
0
# Already started, status return success
$ service script_name status; echo $?
0
# Stop the service, return success
$ service script_name stop; echo $?
0
# Service is stopped, status return failure
$ service script_name status; echo $?
1

```

Stop the service even if the service is already stopped: 가

```

# Start the service, return success
$ service script_name start; echo $?
0
# Stop the service, return success
$ service script_name stop; echo $?
0
# Stop the service again, return success
$ service script_name stop; echo $?
0

```

Report error status when the process exit abnormally: 가 error

```

# Stop the service, return success
$ service script_name start; echo $?
0
# Already started, status return success

```

```
$ service script_name status; echo $?
0
# Kill the process
$ killall process_name
# The process was killed, status return failure
$ service script_name status; echo $?
1
# Stop the service even if it exited abnormally, return success
$ service script_name stop; echo $?
0
```

From: <https://at1.kr/dokuwiki/> - AllThatLinux!

Permanent link: https://at1.kr/dokuwiki/doku.php/%EB%A6%AC%EC%86%8C%EC%8A%A4_%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8_%EC%9E%91%EC%84%B1%EB%B2%95

Last update: 2015/12/07 03:36

