목차

service script 작성방법 3
service scripts 간략 테스트 방법 5

Last update: 2015/12/07 03:36	리소스스크립트작성법	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

참고 URL : https://access.redhat.com/solutions/5912

service script 작성방법

서비스 스크립트는 기본적으로 3가지 기능이 포함되어 있어야 합니다.

1. service 이름으로 start 서비스가 제공되어야 합니다. 실패한 경우에는(**0이 아닌 다른 값**) 성공한 경우에는 **0** 을 반환해야 합니다.
2. service 이름으로 stop 서비스가 제공되어야 합니다. 실패한 경우에는(**0이 아닌 다른 값**) 성공한 경우에는 **0** 을 반환해야 합니다.
3. service 이름으로 status 서비스가 제공되어야 합니다. 실패한 경우에는(**0이 아닌 다른 값**) 성공한 경우에는 **0** 을 반환해야 합니다.

script 문법예

```
#!/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 specfic code here >>>
        /path/to/my-service-start
        exit $?
        ;;
    'stop')
        clulog -s $LOG_INFO "Running user stop script for service $0"
        # <<< Add service stop specfic code here >>>
        /path/to/my-service-stop
        exit $?
        ;;
    'status')
        clulog -s $LOG_INFO "Running user status script for service $0"
        # <<< Add service status specfic 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 간략 테스트 방법

A normal start-status-stop cycle: 정상 가동 상태 start-status-stop 사이클

```
# Service not yet start, status return failure  // 서비스가 아직 시작되지 않은 경우에는 status 값을 failure 로 리턴해야 합니다. ( 0 이 아닌 다른값)
$ 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:만약 서비스가 이미 중지 되어있어도 서비스를 중지시켜라.

```
# 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: 프로세스가 비정상적으로 종료될때 error 상태를 레포트 하세요

```
# 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://atl.kr/dokuwiki/ - **AllThatLinux!**

Permanent link:
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

Last update: **2015/12/07 03:36**