vServer / Virtual Dedicated Server

Automatic network configuration for root servers after reboot

Table of Contents

Explanation

With the help of a cron job, tasks can be executed automatically on a system.

Each cron job consists of three components:

- Script, which is to be executed
- Command, which executes the script regularly
- Action or output of the script

In our example, we create a cron job that automatically loads the network configuration for root servers (VDServer) on restart.

Install required programs

The following commands are used to install the required package.

Debian / Ubuntu

apt-get update & apt-get -y upgrade
apt-get install cron

Fedora / Centos

yum -y update
yum install vixie-cron

Create a cron job

To create a cron job, the file /etc/crontab must be modified. For this purpose, the file can be called via editor or alternatively via the command "crontab -e".

When you call this file for the first time, you will be asked for an editor that you want to use for the adjustments. Here, nano is usually already preselected. You can use an editor of your choice here and select it by entering the corresponding number.

crontab -e

select an editor. To change later, run 'select-editor'.
1. /bin/nano <---- easiest
2. /usr/bin/vim.tiny
3. /bin/ed

Choose 1-3 [1]:

The file /etc/crontab

```
 GNU nano 4.8 /tmp/crontab.cyfsyk/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/home/Tim/Skript/IP_Change
# /home/Tim/Skript/IP_Change definiert den Speicherort des Skripts

# m h dom mon dow user  command
 17 *  *  *  *  root  cd / && run-parts --report /etc/cron.hourly
 25 6  *  *  *  root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
 47 6  *  *  7  root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
 52 6  1  *  *  root  test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )Datei
```

Explanation of the line # m h dom mon dow user command

: defines the line as a comment
m : specifies the minute in which a cron job should be executed (0-59)
h : specifies the hour in which a cron job should be executed (0-23)
dom (day of month) : specifies the day of the month (1-31)
mon (month) : specifies the month (1-12)
dow (day of week) : specifies the day of the week. (0-7) Please note that the numbers 0 and 7 represent Sunday. Monday would be 1 here.
* : any

Example:

```
17 * * * * root cd / && run-parts --report /etc/cron.hourly
```

Here the cron job is executed every minute 17. Since no day, month and hour are specified here, the cron job is executed every minute 17.

Solution without script

The following assumes that a copy of a working network configuration has been created under the following sample path:

/home/user/IP_Backup/IP_Konfiguration_Backup

For use without a script, the following line is added to the /etc/crontab file.

```
@reboot root sleep 15 && cat /home/user/IP_Backup/IP_Konfiguration_Backup >> /etc/network/interfaces/eth0
```

Note: Please check the designation of your network interface in advance. In our example we use eth0.

Following the change, the network service must be restarted. Here, depending on your system, select one of the following commands:

```
@reboot root sleep 25 && service network restart
```

```
@reboot root sleep 25 && systemctl restart network
```

Solution by script

The following assumes that a copy of a working network configuration has been created under the following sample path:

/home/user/IP_Backup/IP_Konfiguration_Backup

It is also assumed that the following script was created under /home/user/Script/IP_Change.

The IP_Change script

```
#!/bin/bash
```

cat /home/user/IP_Backup/IP_Konfiguration_Backup >> /etc/network/interfaces/eth0

#Depending on the system used, the network must be restarted using the service or systemctl command. Please delete the comment sign in front of the correct command:
#service network restart
#systemctl restart network

Now the script must be made executable:

chmod +x /home/user/Skript/IP_Change

Create cronjob for the execution of the script

To run the previously created script via a cronjob at every restart, the following line is added to the /etc/crontab file.

#Runs the script
@reboot root sleep 15 && /home/user/Skript/IP_Change

After successful setup, the script will be executed at every server restart.

Any error message is sent to the root user's mailbox on the system.
Also, these are stored within the log files /var/log/syslog and /var/log/messages respectively.

Unique solution ID: #1401
Author: Bettina Brauer
Last update: 2021-07-18 07:36