Red Hat

Atova TECHNOLOGY

Welcome to

Red Hat Enterprise Linux V9

Online Class

Scheduling

The software utility **cron** also known as **cron job** is a time-based job scheduler in Unix-like computer operating systems.

A **cron job** is a Linux command used for scheduling tasks to be executed sometime in the future.

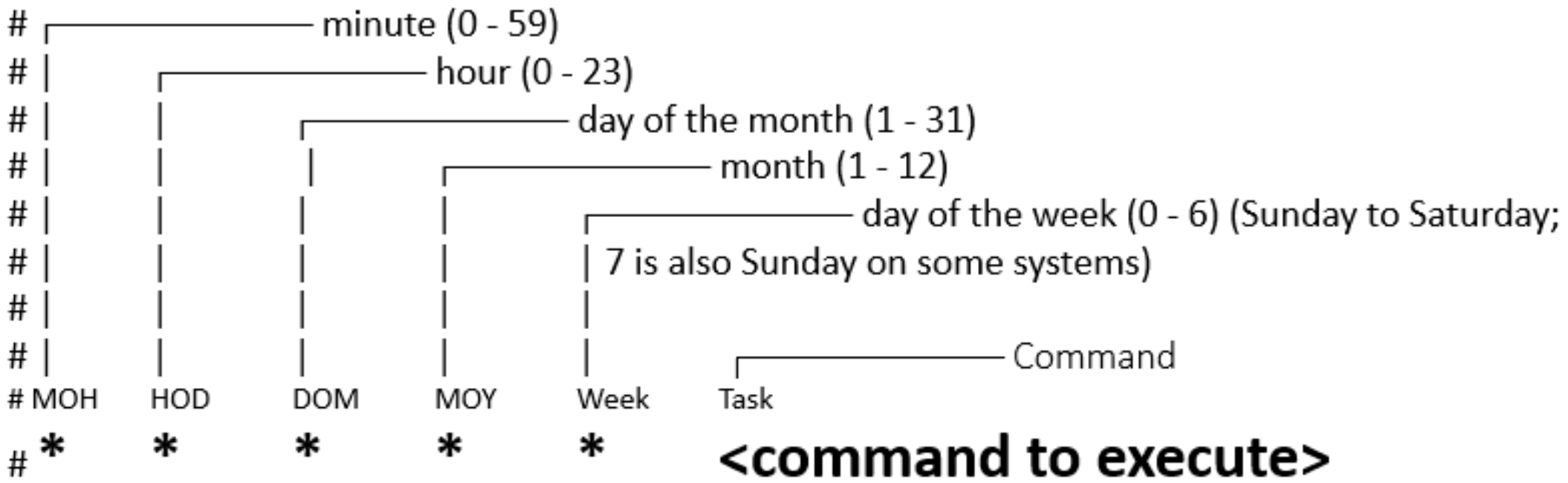
Scheduling

The table below summarizes possible values for the fields and the example syntax:

Field	Possible Values	Syntax	Description
[a] – Minute	0 – 59	7 * * * *	The cron job is initiated every time the system clock shows 7 in the minute's position.
[b] – Hour	0 – 23	0 7 * * *	The cron job runs any time the system clock shows 7am (7pm would be coded as 19).
[c] – Day	0 – 31	0 0 7 * *	The day of the month is 7 which means that the job runs every 7th day of the month.
[d] – Month	0 = none and 12 = December	0 0 0 7 *	The numerical month is 7 which determines that the job runs only in July.
[e] – Day of the Week	0 = Sunday and 7 = Sunday	0 0 * * 7	7 in the current position means that the job would only run on Sundays.

Scheduling

The actions of cron are driven by a **crontab** (cron table) file, a configuration file that specifies shell commands to run periodically on a given schedule.



Scheduling

Each user, including root, can have a cron file. These files don't exist by default, but can be created in the **/var/spool/cron** directory using the **crontab -e** command

For example, the following clears the Apache error log at one minute past midnight (00:01) every day

```
1 0 * * * printf "" > /var/log/apache/error_log
```

Use the following, if want to check the disk space every 10 minutes.

```
*/10 * * * * /home/maverick/check-disk-space >> /opt/maverick-space
```

List Existing Cron Jobs

 # crontab -l

Scheduling

To run a task every 15 minutes:

```
*/15    *       *       *       *       task
```

Edit crontab for a Different User

```
crontab -u  username -e
```

Want to restrict alex user from creating cron

Put the alex users name into /etc/cron.deny file

AT

at is a command-line utility that allows you to schedule commands to be executed at a particular time. Jobs created with **at** are executed only once.

Install **at** on CentOS and Fedora

```
# yum install at
```

restart **at** service

```
# systemctl restart atd
```

Enable in startup

```
# systemctl enable atd
```

AT

Let’s create a job that will be executed at 9:00 am:

```
# at 09:00
warning: commands will be executed using /bin/sh
at>
```

Enter one or more command you want to execute:

```
at> tar -xf /home/linuxize/file.tar.gz
```

When done entering the commands, press Ctrl-D to exit the prompt and save the job:

```
at> <EOT>
job 4 at Tue May 5 09:00:00 2020
```

AT

Command to list the user's pending jobs:

 # at -l

or

 # atq

Schedule a job for the coming Monday at a time twenty minutes later than the current time:

 # at Monday +20 minutes

To delete a **at** job:

 # atrm job_number

Chrony (NTP- Network Time Protocol)

chrony is an implementation of the Network Time Protocol (NTP). It's a replacement for the ntpd, which is a reference implementation of the NTP.

It's the default NTP client and server in Red Hat Enterprise Linux 8 and available in many Linux distributions

Chrony (NTP- Network Time Protocol)

Chrony comes with two programs:

chronyc – command line interface for chrony

chronyd – is the Service Daemon

Install Chrony in RedHat Enterprise Linux

```
# yum -y install chrony
```

Start and enable **chronyd** service

```
# systemctl start chronyd
#
# systemctl enable chronyd
```

Check Chrony Synchronization

```
# chronyc tracking
```

To check information about chrony's sources

```
# chronyc sources
```

Chrony (NTP- Network Time Protocol)

Atova TECHNOLOGY

Configure Chrony

 # vim /etc/chrony.conf

Use public servers from the pool.ntp.org project.
Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.asia.pool.ntp.org
server 1.asia.pool.ntp.org
server 2.asia.pool.ntp.org
server 3.asia.pool.ntp.org

Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

Allow the system clock to be stepped in the first three updates
if its offset is larger than 1 second.
makestep 1.0 3

Enable kernel synchronization of the real-time clock (RTC).
.....

SSH (Secure Shell)

ssh stands for **“Secure Shell”**. It is a protocol used to securely connect to a remote server/system. ***ssh*** is secure in the sense that it transfers the data in encrypted form between the host and the client. ***ssh*** runs at TCP/IP port 22.

SSH (Secure Shell)

Syntax:

```
ssh user_name@host(IP/Domain_name)
```

Install ***ssh*** package

```
# yum install -y openssh-server
```

Allow ***ssh*** in firewall:

```
# firewall-cmd --permanent --add-service=ssh
#
# firewall-cmd --reload
```

Start & enable ***sshd***

```
# systemctl start sshd
#
# systemctl enable sshd
```

How to secure ssh server

Open ssh configuration file

```
# vim /etc/ssh/sshd_config
```

1. Change default port:

Port ….

2. Disable root login:

PermitRootLogin no

3. Allow Specific user:

AllowUsers validuser1 validuser2

4. Restrict ***ssh*** login to specific IP:

ListenAddress 192.168.1.100

5. Disable password authentication:

PasswordAuthentication no

6. Set Idle Timeout Interval:

ClientAliveInterval 300
ClientAliveCountMax 0

ssh keygen

Generate *ssh* key

 # ssh-keygen

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kbuzdar/.ssh/id_rsa.
Your public key has been saved in /home/kbuzdar/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:4ncQUR6oYqAmmjA2IY+UIlNIdqjlGQKmjH4cDIFhivk kbuzdar@vitux.com
The key's randomart image is:
+---[RSA 3072]----+
|*@=.     .oo     |
|^*B      .o .    |
|%X.*    .. .     |
|O== + .   .      |
|*+E+ .. S        |
|o .   . . .      |
|       . . .     |
|        . .      |
|                 |
+----[SHA256]-----+
```

transfer ssh key to other machine

Atova TECHNOLOGY

ssh-copy-id username@remote_host

Output

The authenticity of host 'remote_host_ip (remote_host_ip)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes

username@remote_hosts's password:

Number of key(s) added: 1
Now try logging into the machine, with: "ssh 'username@remote_host'"
and check to make sure that only the key(s) you wanted were added.

transfer files via *ssh*

Securely copy file

Command Syntax:

 # scp <source> <remote_server_destination>

Example:

 # scp /path/to/file username@remote_server:/path/to/destination

Copy file from one server to another server

 # scp /opt/file root@192.168.10.209:~/