RHCEv8 Online Class 06062021 10:00pm
RHCSA-sysadmin
- Scheduling Future Tasks
- Tuning System Performance
- Accessing Network-Attached Storage

Scheduling Future Tasks
SCHEDULING A DEFERRED USER JOB
DESCRIBING DEFERRED USER TASKS
-Sometimes you might need to run a command, or set of commands, at a set point in the future.
-Scheduled commands are often called tasks or jobs, and the term deferred indicates that these tasks or jobs are going to run in the future.
-The solutions available to RHEL users for scheduling deferred tasks is **at**.
-at is temporary scheduler, it means after run it won’t be accessible. no history

at info
package: at.x86_64
daemon: atd.service

Implement at on linux
```
# yum install at.x86_64 -y
# yum update at.x86_64 -y
# systemctl enable atd.service
# systemctl start atd.service
# systemctl status atd.service
```

work with at
-create
```
# at now +3min
at> mkdir /tmp/testdir
at> touch /tmp/testdir/file1
at> echo "Hello at!" >/tmp/testdir/file1
at> <EOT>                                   press Ctrl+d to save and quit
job 1 at Sun Jun  6 10:21:00 2021
```
-watch
```
# atq
or
# at -l
1       Sun Jun  6 10:21:00 2021 a root
1                                 ->job-id/number
Sun Jun  6 10:21:00 2021          ->timestamp
a                                 ->priority
root                              ->who ordered it
```

```
# at now +5min
at> userad user1
at> echo "redh" | [aasws --stdin user1
at> <EOT>
job 2 at Sun Jun  6 10:28:00 2021
```
-watch
```
# atq
```
-check task’s details
```
# at -c <job-id>
# at -c 2
```
-make correction
```
# ls /var/spool/at/
a00002019cbd2a                              ->its ephemeral file, won’t be accessible after run task
# vim /var/spool/at/a00002019cbd2a
useradd user1
echo "redhat" | passwd --stdin user1
touch /tmp/testdir/file2
:wq
```
-delete task
```
# atrm <job-id>
# atrm 3
```

timestamp
reference file: /usr/share/doc/at/timespec

```
# cat /usr/share/doc/at/timespec
```

ex:

```
# date
Sun Jun  6 10:37:08 IST 2021
# at now +10min
job 4 at Sun Jun  6 10:47:00 2021
# at teatime
job 5 at Sun Jun  6 16:00:00 2021
# at 8:00am 07 june 21
job 6 at Mon Jun  7 08:00:00 2021
# at 11:30pm 30 june 21
job 7 at Wed Jun 30 23:30:00 2021
```

SCHEDULING RECURRING USER JOBS
DESCRIBING RECURRING USER JOBS
-Jobs scheduled to run repeatedly are called **recurring** jobs.
-RHEL systems ship with the crond daemon, provided by the cronie package, enabled and started by default specifically for recurring jobs.

cron info
package: cronie.x86_64, cronie-anacron.x86_64, crontabs.noarch
daemon: crond.service
log: /var/log/cron

Implement cron on linux

```
# yum install cron* -y
# yum update cron* -y
# systemctl enable crond.service
# systemctl start crond.service
# systemctl restart crond.service
```

work with cron
-current user

```
# crontab -e  edit
          -l    list
          -r   remove
          -ir  ask question before remove
```

-remote user

```
# crontab -eu <username>
          -lu <username>
          -r <username>
          -ir <username>
```

How cron works
in cron we have **5 starts**

```
# cat /etc/crontab
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12)
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7)
# |  |  |  |  |
# *  *  *  *  *    <user-name>  <command to be executed>
```

ex:
-run task daily 10:30am
30 10 * * * <command>
-run task 07Jun21 11:30pm
30 23 7 6 1 <command>
-run task monthly 2:45pm
45 14 1 * *

-current user

create

```
# crontab -e
03 11 * * * /usr/sbin/useradd user2
04 11 * * * /usr/bin/echo "redhat" | passwd --stdin user2
:wq!
```

watch

```
# crontab -l
```

cron file stored in:

```
# ls /var/spool/cron/
root
# vim /var/spool/cron/root
08 11 * * * /usr/bin/touch /tmp/testdir/file3
:wq!
# crontab -l
08 11 * * * /usr/bin/touch /tmp/testdir/file3
```

-delete

```
# crontab -ir
crontab: really delete root's crontab? yes/no
# crontab -r
```

how to block users access to cron command?

```
# su - devops
$ crontab -e                    ->it works
$ exit
#
now:
# vim /etc/cron.deny
devops
:wq!
# su - devops
$ crontab -e                    ->it won't works
You (devops) are not allowed to use this program (crontab)
```

-remote user

```
# crontab -eu devops
# crontab -lu devops
17 11 * * * /usr/bin/touch /home/devops/file1.txt
# ls /var/spool/cron/
devops
# cat /var/spool/cron/devops
17 11 * * * /usr/bin/touch /home/devops/file1.txt
# ll /home/devops/
total 0
-rw-r--r--. 1 devops devops 0 Jun  6 11:17 file1.txt
# crontab -iru devops
crontab: really delete devops's crontab? yes/no
# crontab -ru devops
```

Accessing Network-Attached Storage
File-based Storage Protocols/Network Attach Storage-NAS
1-Network File Sharing-NFS
2-SAMBA/CIFS Common Internet File Sharing-CIFS
3-File Transfer Protocol-FTP

Block-based Storage Protocol/Storage Area Network-SAN
1-iSCSI

MOUNTING NETWORK-ATTACHED STORAGE WITH NFS
network file sharing/system its internet standard protocol created by sun microsystems in 1984 to share unix to unix storage.

NFS Info
package: nfs-utils.x86_64, rpcbind.x86_64
daemon: nfs-server.service
config file: /etc/exports, /etc/nfs.conf
port: 2049/tcp
log: /var/log/message

Implement NFS on linux
```
servera.lab.example.com   172.25.250.10     nfs-server
serverb.lab.example.com   172.25.250.11     nfs-client
```

ex:
create 1gb lvm partition, export it through NFS
server-side
servera
```
# cat /proc/partitions
# fdisk /dev/sdb
p
n
e
Partition number (1-4, default 1): Enter
First sector: Enter
Last sector: Enter
p
/dev/sdb1       2048 10485759 10483712   5G  5 Extended
n
Adding logical partition 5
Enter
+2G
p
/dev/sdb1       2048 10485759 10483712   5G  5 Extended
/dev/sdb5       4096  4198399  4194304    2G 83 Linux
t
Partition number (1,5, default 5): 5
Hex code (type L to list all codes): 8e
p
/dev/sdb1       2048 10485759 10483712   5G  5 Extended
/dev/sdb5       4096  4198399  4194304    2G 8e Linux LVM
w
# udevadm settle
# fdisk -l /dev/sdb
# pvcreate /dev/sdb5
# pvs
/dev/sdb5     lvm2 ---    2.00g 2.00g
# vgcreate vg1 /dev/sdb5
# vgs
vg1   1   0   0 wz--n-  <2.00g <2.00g
# lvcreate -n lv1 -L +1G vg1
lv1  vg1 -wi-a-----   1.00g
# mkfs.xfs /dev/mapper/vg1-lv1
# mkdir /mnt/lv1
# blkid
/dev/mapper/vg1-lv1: UUID="760fefb3-f803-42f3-bde6-afdd9bab4782" TYPE="xfs"
# echo "/dev/mapper/vg1-lv1 /mnt/lv1 xfs defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
/dev/mapper/vg1-lv1 xfs     1014M   40M  975M   4% /mnt/lv1
# ls -ld /mnt/lv1/
drwxr-xr-x. 2 root root 6 Jun  6 12:07 /mnt/lv1/
# chmod 757 /mnt/lv1/
# ls -ld /mnt/lv1/
drwxr-xrwx. 2 root root 6 Jun  6 12:07 /mnt/lv1/
# yum list nfs* rpc*
# yum install nfs* rpc* -y
# systemctl enable nfs-server.service
# systemctl start nfs-server.service
# systemctl status nfs-server.service
# firewall-cmd --permanent --add-service={nfs,mountd,rpc-bind}
# firewall-cmd --reload
# firewall-cmd --list-all
# vim /etc/exports
<nfs export directory> <to whom>:(permission)
/mnt/lv1 *(rw,sync)
/mnt/lv1 *.lab.example.com(ro,sync)
/mnt/lv1 172.25.250.10/24(rw,sync)
/mnt/lv1 172.25.0.0/16(ro,sync)

/mnt/lv1 172.25.250.0/24(rw,sync)
:wq
# systemctl restart nfs-server.service
```

local

```
# exportfs -rva
exporting 172.25.250.0/24:/mnt/lv1
```

global

```
# showmount -e 172.25.250.10
Export list for 172.25.250.10:
/mnt/lv1 172.25.250.0/24
# touch /mnt/lv1/file1
# ls -l /mnt/lv1/file1
-rw-r--r--. 1 root root 6 Jun  6 12:30 /mnt/lv1/file1
# chmod 646 /mnt/lv1/file1
# ls -l /mnt/lv1/file1
-rw-r--rw-. 1 root root 13 Jun  6 12:32 /mnt/lv1/file1
```

client-side

serverb

```
# yum install nfs-utils.x86_64 -y
# showmount -e 172.25.250.10
# showmount -e 172.25.250.10
Export list for 172.25.250.10:
/mnt/lv1 172.25.250.0/24
# mkdir /mnt/nfs
```

mount nfs

1-temporary

```
# mount -t nfs -o ro,sync 172.25.250.10:/mnt/lv1 /mnt/nfs
# mount -a
# df -hT
172.25.250.10:/mnt/lv1   nfs4     1014M   39M  975M   4% /mnt/nfs
# touch /mnt/nfs/file1
touch: cannot touch '/mnt/nfs/file1': Read-only file system
# umount /mnt/nfs
# mount -t nfs -o rw,sync 172.25.250.10:/mnt/lv1 /mnt/nfs
# mount -a
# echo "severb" >>/mnt/nfs/file1
```

2-prsistently

```
showmount -e 172.25.250.10
Export list for 172.25.250.10:
/mnt/lv1 172.25.250.0/24
# echo "172.25.250.10:/mnt/lv1 /mnt/nfs nfs defaults 0 0" >>/etc/fstab
or
# echo "servera.lab.example.com:/mnt/lv1 /mnt/nfs nfs defaults 0 0" >>/etc/fstab
# mount -a
# df -hT
172.25.250.10:/mnt/lv1 nfs4     1014M   39M  975M   4% /mnt/nfs

# umount /mnt/nfs
# vim /etc/fstab
remove nfs record
:wq!
# mount -a
# df -hT
```

3-autofs

```
# yum install autofs -y
# systemctl enable autofs.service
# systemctl start autofs.service
# systemctl status autofs.service
```

autofs has 2 main config files

1-/etc/auto.master	->define mount point
2- /etc/auto.misc	->define mount device info

```
# showmount -e 172.25.250.10
Export list for 172.25.250.10:
/mnt/lv1 172.25.250.0/24
# vim /etc/auto.master
/mnt/nfs        /etc/auto.misc
# vim /etc/auto.misc
coss          -fstype=nfs,rw,sync,vers=4.0    172.25.250.10:/mnt/lv1
:wq!
# systemctl restart autofs.service
# mount -a
# cd /mnt/nfs/coss
# ls
file1  file2
# df -hT
172.25.250.10:/mnt/lv1 nfs4     1014M   39M  975M   4% /mnt/nfs/coss
# tail /etc/mtab
/etc/auto.misc /mnt/nfs autofs rw,relatime,fd=18,pgrp=26290,timeout=300,minproto=5,maxproto=5,indirect,pipe_ino=81501  0  0
172.25.250.10:/mnt/lv1 /mnt/nfs/coss nfs4 rw,sync,relatime,vers=4.0,rsize=524288,wsize=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,clientaddr=172.25.250.11,local_lock=none,addr=172.25.250.10  0  0
```

Tuning System Performance
ADJUSTING TUNING PROFILES
TUNING SYSTEMS
-System administrators can optimize the performance of a system by adjusting various device settings based on a variety of use case workloads.
-The **tuned daemon** applies tuning adjustments both statically and dynamically, using tuning profiles that reflect particular workload requirements.

tuned Info
package: tuned.noarch
daemon: tuned.service

Implement tuned on linux
```
# yum install tuned.noarch -y
# yum update tuned.noarch -y
# systemctl enable tuned.service
# systemctl start tuned.service
# systemctl restart tuned.service
# tuned-adm
active     list      off       profile    recommend   verify
off           ->tuned goes off
list          ->list of profiles
profile    ->select profile to active
active     ->what is active profile now
recommend   ->recommend profile
verify      ->verify active profile

# tuned-adm list
# tuned-adm active
Current active profile: virtual-guest
# tuned-adm profile powersave
# tuned-adm active
Current active profile: powersave
# tuned-adm verify
Verification failed
# tuned-adm recommend
virtual-guest
# tuned-adm profile virtual-guest
# tuned-adm active
Current active profile: virtual-guest
# tuned-adm verify
Verfication succeeded
```