Using Shell Scripting to Automate Linux System Maintenance Tasks – Part 4

Gabriel Cánepa | Last Updated: August 14, 2015 | Read Time: 4 mins | Linux Certifications, RHCE | 27 Comments

Some time ago I read that one of the distinguishing characteristics of an effective system administrator / engineer is laziness. It seemed a little contradictory at first but the author then proceeded to explain why:

RHCE Series: Automate Linux System Maintenance Tasks – Part 4

if a sysadmin spends most of his time solving issues and doing repetitive tasks, you can suspect he or she is not doing things quite right. In other words, an effective system administrator / engineer should develop a plan to perform repetitive tasks with as less action on his / her part as possible, and should foresee problems by using,

for example, the tools reviewed in Part 3 – Monitor System Activity Reports Using Linux Toolsets of this series. Thus, although he or she may not seem to be doing much, it's because most of his / her responsibilities have been taken care of with the help of shell scripting, which is what we're going to talk about in this tutorial.

What is a shell script?

In few words, a shell script is nothing more and nothing less than a program that is executed step by step by a shell, which is another program that provides an interface layer between the Linux kernel and the end user.

By default, the shell used for user accounts in RHEL 7 is bash (**/bin/bash**). If you want a detailed description and some historical background, you can refer to this Wikipedia article.

To find out more about the enormous set of features provided by this shell, you may want to check out its **man page**, which is downloaded in in PDF format at (Bash Commands). Other than that, it is assumed that you are familiar with Linux commands (if not, I strongly advise you to go through A Guide from Newbies to SysAdmin article in **Tecmint.com** before proceeding). Now let's get started.

Writing a script to display system information

For our convenience, let's create a directory to store our shell scripts:

```
# mkdir scripts
# cd scripts
```

And open a new text file named `system_info.sh` with your preferred text editor. We will begin by inserting a few comments at the top and some commands afterwards:

```
#!/bin/bash

# Sample script written for Part 4 of the RHCE series
# This script will return the following set of system information:
# -Hostname information:
echo -e "\e[31;43m***** HOSTNAME INFORMATION *****\e[0m"
hostnamectl
echo ""
```

```
# -File system disk space usage:
echo -e "\e[31;43m***** FILE SYSTEM DISK SPACE USAGE *****\e[0m"
df -h
echo ""
# -Free and used memory in the system:
echo -e "\e[31;43m ***** FREE AND USED MEMORY *****\e[0m"
free
echo ""
# -System uptime and load:
echo -e "\e[31;43m***** SYSTEM UPTIME AND LOAD *****\e[0m"
uptime
echo ""
# -Logged-in users:
echo -e "\e[31;43m***** CURRENTLY LOGGED-IN USERS *****\e[0m"
who
echo ""
# -Top 5 processes as far as memory usage is concerned
echo -e "\e[31;43m***** TOP 5 MEMORY-CONSUMING PROCESSES *****\e[0m"
ps -eo %mem,%cpu,comm --sort=-%mem | head -n 6
echo ""
echo -e "\e[1;32mDone.\e[0m"
```

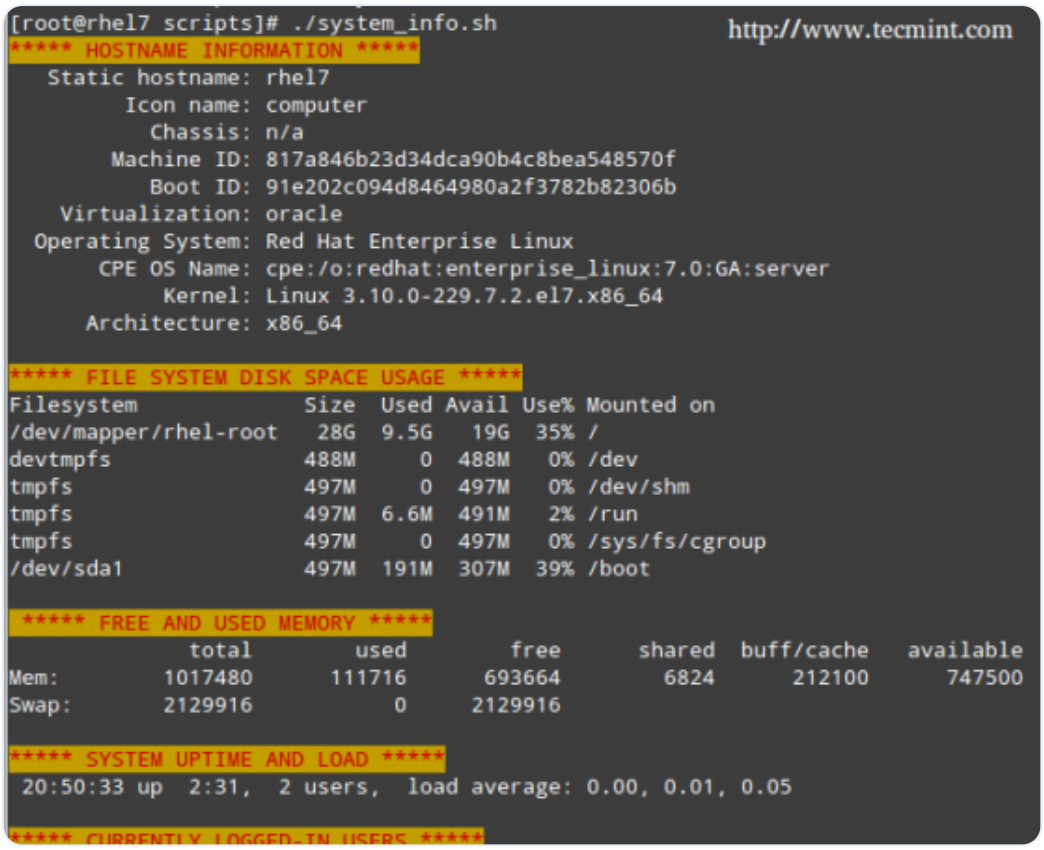
Next, give the script execute permissions:

```
# chmod +x system_info.sh
```

and run it:

```
./system_info.sh
```

Note that the headers of each section are shown in color for better visualization:



Server Monitoring Shell Script

That functionality is provided by this command:

```
echo -e "\e[COLOR1;COLOR2m<YOUR TEXT HERE>\e[0m"
```

Where COLOR1 and COLOR2 are the foreground and background colors, respectively (more info and options are explained in this entry from the Arch Linux Wiki) and <YOUR TEXT HERE> is the string that you want to show in color.

Automating Tasks

The tasks that you may need to automate may vary from case to case. Thus, we cannot possibly cover all of the possible scenarios in a single article, but we will present three classic tasks that can be automated using shell scripting:

1) update the local file database, 2) find (and alternatively delete) files with 777 permissions, and 3) alert when filesystem usage surpasses a defined limit.

Let's create a file named `auto_tasks.sh` in our scripts directory with the following content:

```
#!/bin/bash

# Sample script to automate tasks:
# -Update local file database:
echo -e "\e[4;32mUPDATING LOCAL FILE DATABASE\e[0m"
updatedb
if [ $? == 0 ]; then
        echo "The local file database was updated correctly."
else
        echo "The local file database was not updated correctly."
fi
echo ""

# -Find and / or delete files with 777 permissions.
echo -e "\e[4;32mLOOKING FOR FILES WITH 777 PERMISSIONS\e[0m"
# Enable either option (comment out the other line), but not both.
# Option 1: Delete files without prompting for confirmation. Assumes GNU ve
#find -type f -perm 0777 -delete
# Option 2: Ask for confirmation before deleting files. More portable acros
find -type f -perm 0777 -exec rm -i {} +;
echo ""
# -Alert when file system usage surpasses a defined limit
echo -e "\e[4;32mCHECKING FILE SYSTEM USAGE\e[0m"
THRESHOLD=30
while read line; do
        # This variable stores the file system path as a string
        FILESYSTEM=$(echo $line | awk '{print $1}')
        # This variable stores the use percentage (XX%)
        PERCENTAGE=$(echo $line | awk '{print $5}')
        # Use percentage without the % sign.
        USAGE=${PERCENTAGE%?}
        if [ $USAGE -gt $THRESHOLD ]; then
                echo "The remaining available space in $FILESYSTEM is criti
        fi
done < <(df -h --total | grep -vi filesystem)
```

Please note that there is a space between the two `<` signs in the last line of the script.

```
[root@rhel7 scripts]# ./auto_tasks.sh
UPDATING LOCAL FILE DATABASE
The local file database was updated correctly.

LOOKING FOR FILES WITH 777 PERMISSIONS
rm: remove regular empty file './file.txt'? y

CHECKING FILE SYSTEM USAGE
The remaining available space in /dev/mapper/rhel-root is critically low. Used: 35%
The remaining available space in /dev/sda1 is critically low. Used: 39%
The remaining available space in total is critically low. Used: 33%
[root@rhel7 scripts]#
```

Shell Script to Find 777 Permissions

Using Cron

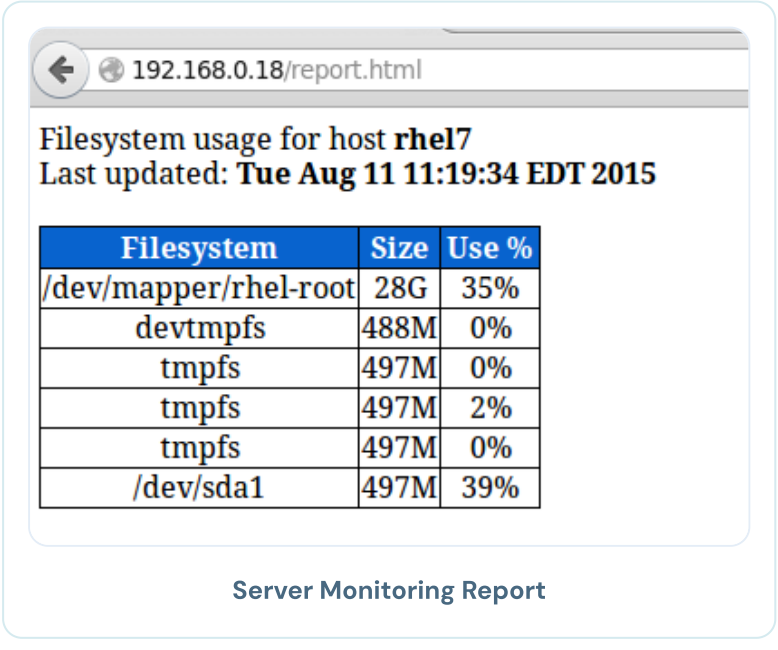
To take efficiency one step further, you will not want to sit in front of your computer and run those scripts manually. Rather, you will use **cron** to schedule those tasks to run on a periodic basis and sends the results to a predefined list of recipients via email or save them to a file that can be viewed using a web browser.

The following script (filesystem_usage.sh) will run the well-known **df -h** command, format the output into a HTML table and save it in the **report.html** file:

```
#!/bin/bash
# Sample script to demonstrate the creation of an HTML report using shell s
# Web directory
WEB_DIR=/var/www/html
# A little CSS and table layout to make the report look a little nicer
echo "<HTML>
<HEAD>
<style>
.titulo{font-size: 1em; color: white; background:#0863CE; padding: 0.1em 0.
table
{
border-collapse:collapse;
}
table, td, th
{
border:1px solid black;
}
</style>
```

```
<meta http-equiv='Content-Type' content='text/html; charset=UTF-8' />
</HEAD>
<BODY>" > $WEB_DIR/report.html
# View hostname and insert it at the top of the html body
HOST=$(hostname)
echo "Filesystem usage for host <strong>$HOST</strong><br>
Last updated: <strong>$(date)</strong><br><br>
<table border='1'>
<tr><th class='titulo'>Filesystem</td>
<th class='titulo'>Size</td>
<th class='titulo'>Use %</td>
</tr>" >> $WEB_DIR/report.html
# Read the output of df -h line by line
while read line; do
echo "<tr><td align='center'>" >> $WEB_DIR/report.html
echo $line | awk '{print $1}' >> $WEB_DIR/report.html
echo "</td><td align='center'>" >> $WEB_DIR/report.html
echo $line | awk '{print $2}' >> $WEB_DIR/report.html
echo "</td><td align='center'>" >> $WEB_DIR/report.html
echo $line | awk '{print $5}' >> $WEB_DIR/report.html
echo "</td></tr>" >> $WEB_DIR/report.html
done < <(df -h | grep -vi filesystem)
echo "</table></BODY></HTML>" >> $WEB_DIR/report.html
```

In our RHEL 7 server (192.168.0.18), this looks as follows:



Server Monitoring Report

You can add to that report as much information as you want. To run the script every day at 1:30 pm, add the following crontab entry:

```
30 13 * * * /root/scripts/filesystem_usage.sh
```

Summary

You will most likely think of several other tasks that you want or need to automate; as you can see, using shell scripting will greatly simplify this effort. Feel free to let us know if you find this article helpful and don't hesitate to add your own ideas or comments via the form below.

RHCE Certification

Hey TecMint readers,

Exciting news! Every month, our top blog commenters will have the chance to win fantastic rewards, like free Linux eBooks such as RHCE, RHCSA, LFCS, Learn Linux, and Awk, each worth $20!

Learn more about the contest and stand a chance to win by sharing your thoughts below!

PREVIOUS ARTICLE:

How to Deploy Virtual Machines in RHEV Environment – Part 4

NEXT ARTICLE:

Happy 3rd Birthday to TecMint Community

Gabriel Cánepa

Gabriel Cánepa is a GNU/Linux sysadmin and web developer from Villa Mercedes, San Luis, Argentina. He works for a worldwide leading consumer product company and takes great pleasure in using FOSS tools to increase productivity in all areas of his daily work.

Each tutorial at TecMint is created by a team of experienced Linux system administrators so that it meets our high-quality *standards.*

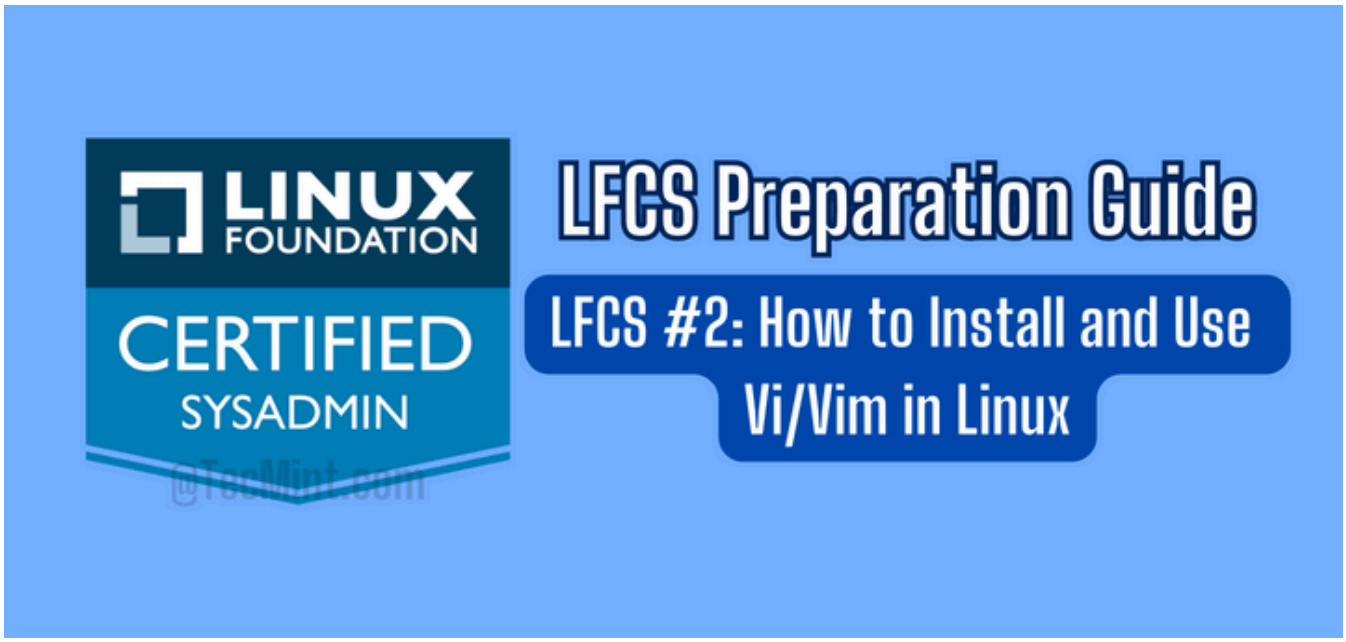
Related Posts

LFCS #6: How to Assemble Partitions as RAID Devices and Create System Backups

LFCS #5: How to Mount Local and Network (Samba & NFS) Filesystems in Linux

LFCS #4: How to Partition Storage Devices in Linux

LFCS #3: How to Archive Files, Set File Permissions and Finding Files in Linux



LFCS #2: How to Install and Use Vi/Vim as a Full Text Editor in Linux

LFCS #1: How to Use 'sed' Command to Create, Edit, and Manipulate Files in Linux

27 Comments

Leave a Reply

Sunil Darna

August 29, 2019 at 1:58 pm

I'm trying to execute below script to get Memory information for clients. But It throwing error with " bash: Total: command not found.

bash: line 1: Used: command not found
bash: line 2: Free: command not found

I understood that shell is confused to execute "**invited quotes**". But I used `" ` "` symbol also same output. Please help me to execute with desired output.

```
#!/bin/sh

echo "MEMORY"

cmds="free -t -m | grep 'Mem' | awk '{ print "Total : "$2 " MB";print "Used : "$3" MB";print "Free : "$4" MB";}' "

for ip in `cat /home/server.txt`
do
ssh -i "/home/ec2-user/Office_Laptop_key.pem" ec2-user@${ip} "$cmds"|tr '\n' ','|sed 's/,,/,NULL,/g'
done
```

Reply

Sajal
May 25, 2019 at 10:43 pm

How can run this report to remote servers to gather the data and display for individual servers in one report.

Reply

Bile

September 2, 2018 at 3:44 pm

With the help of this script I can show system information, thanks.

Please post more related things like install lamp, network profiles and lots of more....

Reply

Admin

Ravi Saive

September 3, 2018 at 10:16 am

@Bile,

Just do a little search on Tecmint.com, you will get all articles about LAMP, LEMP and network related stuff..

Reply

Garima Jain

March 19, 2018 at 11:40 am

Is there a way to execute powershell script from the shell script? (Calling ps1 from .sh file in centos)

Reply

Admin

Ravi Saive

March 19, 2018 at 4:18 pm

@Garima,

I never worked with powershell, so don't have idea it works or not, but you could check this thread for solution – https://stackoverflow.com/questions/35865408/call-powershell-script-from-unix-shell-script

Reply

Softsuit

March 14, 2018 at 12:46 am

Great scripts intro. The data provided helps monitor usage. Thanks for your professional help.

Respectfully

Reply

buntu

December 23, 2016 at 9:03 pm

Its really helpful and i found what actually I am looking for from past one day. Hope now i will do my assignment because my confusions are now clear

Thanks you so much sir

Reply

Gabriel A. Cánepa

January 2, 2017 at 8:24 pm

That is great to hear! We're glad that you found this article useful.

Reply

Keenan Lawrence

September 26, 2016 at 2:04 pm

Such a brilliant article! Clear and concise. I love the reporting bit at the end. Thank you for taking the time to write this :)

Reply

Gabriel A. Cánepa

September 27, 2016 at 8:23 pm

Thank you for such a kind comment. We're glad you found it useful.

Reply

Raj

July 21, 2016 at 12:31 pm

I was asked to capture system information(s)and provide the output in excel sheet using Shell script which has to be automated. While browsing I came across this site and impressed with the way it was explained. How much would be the fee for preparing the script.

Thanks

Reply

Admin

Ravi Saive

July 21, 2016 at 12:32 pm

@Raj,

Thanks for finding this site useful, for any custom services you can contact us at admin@tecmint.com.

Reply

Heena

July 19, 2016 at 4:42 pm

When I am trying to add done < <(`) i am getting below error`

```
filesystem.sh: line 10: syntax error near unexpected token `<'
filesystem.sh: line 10: `done < <(df -h | grep -vi filesystem)'

Can you please help me out here?

Also, I am trying to run some scripts when my OS boots. I added the files in rc.local as well as /etc/init.d/ but it didnt help. I wanted to write some welcome script etc but it isnt working. Can you advice me? That will really help.

Thanks
```

Reply

Việt Adm

July 25, 2016 at 10:43 am

@Heena
I also encountered the same error as you. To fix that, I did the following:

df –h | while read line; do
echo “” >> $WEB_DIR/report.html
….
echo “” >> $WEB_DIR/report.html
done

It has worked well.

Reply

Việt Adm

July 25, 2016 at 10:50 am

I have also encountered the same error as you. I've fixed it like this:

df -h | while read line; do
echo "" >> $WEB_DIR/report.html
done

I'm sure it works.
By the way, I'm using CentOS.

Reply

Admin

Ravi Saive
July 25, 2016 at 11:09 am

@Viet,

Thanks for the fix, hope it will help other CentOS users..

Reply

ankit
February 21, 2020 at 6:15 pm

I am also using centos and facing the same error and I have tried this:

```
df -h | while read line; do
echo "" >> $WEB_DIR/report.html
done
```

Also but got an error...

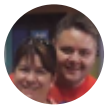
Reply

Andres

June 17, 2016 at 2:44 am

Very good article , Gabriel. I'm starting in this topic to automate processes linux, and I found Super to start thinking about other tasks.

Reply



Gabriel Cánepa

June 21, 2016 at 8:26 am

@Andres,
Thank you for your kind words, and for taking the time to comment on this post.
Feel free to share!

Reply

Pavan

May 28, 2016 at 2:04 pm

sir i want to add user into system without password is it possible to write shell script for that...plz help for this topic

Reply

Gabriel A. Cánepa

May 30, 2016 at 5:56 pm

@Pavan,
I am a little unclear as to why anyone would want to do that. Please share the context of your request and we will be more than happy to help.

Reply

poorva

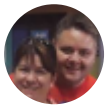
March 30, 2016 at 3:03 pm

Please automate the below steps using shell script.

[root@node1fs ~]# /etc/init.d/infrastructure_manager stop
Stopping Infrastructure Manager Daemon [OK]
[root@node1fs ~]# rpm -qa | grep object
fm-objectshare-services-component-2.1-SNAPSHOT20160323191639.noarch
[root@node1fs ~]# rpm -e fm-objectshare-services-component-2.1-SNAPSHOT20160323191639.noarch
[root@node1fs ~]# rpm -ivh fm-objectshare-services-component-2.1-SNAPSHOT20160330045003.noarch.rpm
Preparing...
[100%]
Installing HTTP
1:fm-objectshare-services### [100%]
[root@node1fs ~]# /etc/init.d/infrastructure_manager start
Starting Infrastructure Manager Daemon [OK]
waiting for PML start

waiting for PML start
waiting for PML start
waiting for PML start
Infrastructure Manager Daemon started.
[root@node1fs ~]#

Reply



Gabriel A. Cánepa
March 30, 2016 at 5:55 pm

@poorva,
We are currently offering support for cases like these. For a minimum fee, either Ravi Saive (the owner of Tecmint) or I can create a shell script that automates this process and creates a nice looking report. Feel free to contact us for a quotation if you want. We can answer questions related to this article for free as an added value, but for tasks that require further work we need to charge people a fee.

Reply

vicotr
March 29, 2016 at 3:50 pm

this did not help

Reply

Alican
March 22, 2016 at 2:56 pm

this did not help

Reply

Gabriel A. Cánepa
March 22, 2016 at 5:33 pm

@Alican,
Would you mind very much explaining why “this did not help”? What were your expectations when you started reading this article? We would be happy to consider your suggestions to improve it.

Reply

Got Something to Say? Join the Discussion...

Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our comment policy to ensure a respectful and constructive conversation.

Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.

Name *

Email *

Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search...

Do You Enjoy My Blog?

Support from readers like YOU keeps this blog running. Buying me a cup of coffee is a simple and affordable way to show your appreciation and help keep the posts coming!

Buy Me a Coffee

Linux Commands and Tools

How to Use 'cat' and 'tac' Commands with Examples in Linux

How to Use 'dir' Command with Different Options and Arguments in Linux

cloc – Count Lines of Code in Many Programming Languages

8 Netcat (nc) Command with Examples

Cron Vs Anacron: How to Schedule Jobs Using Anacron on Linux

How to Use Chown Command to Change File Ownership [11 Examples]

Linux Server Monitoring Tools

20 Useful Commands of 'Sysstat' Utilities (mpstat, pidstat, iostat and sar) for Linux Performance Monitoring

Suricata – A Intrusion Detection, Prevention, and Security Tool

Netdata – A Real-Time Performance Monitoring Tool for Linux Systems

Swatchdog – Simple Log File Watcher in Real-Time in Linux

How to Install Nagios Monitoring in RHEL, Rocky, and AlmaLinux

Duf – A Better Linux Disk Monitoring Utility

Learn Linux Tricks & Tips

How To Assign Output of a Linux Command to a Variable

4 Ways to Find Server Public IP Address in Linux Terminal

How to Find and Sort Files Based on Modification Date and Time in Linux

How to Delete Root Mails (Mailbox) File in Linux

How to Check Bad Sectors or Bad Blocks on Hard Disk in Linux

How to Clone a Partition or Hard drive in Linux

Best Linux Tools

Top 6 Command Line Music Players for Linux Users

5 Best Open-Source eLearning Platforms for Linux in 2024

5 Open Source Log Monitoring and Management Tools for Linux

23 Best Open Source Text Editors for Linux in 2024

21 Best Slack Alternatives for Team Chat [Free & Paid]

5 CLI Tools for Downloading Files and Browsing Internet in Terminal