

IBM LinuxONE Bootcamp

Linux usage lab

Last updated October 17th, 2025

Table of Contents

Introduction.....	3
Lab Environment	3
Lab Objectives.....	3
Lab Access.....	3
System information commands	5
User and group management commands	11
File management commands.....	13
Network commands.....	19
Package management commands	20
References	24
Contact.....	24

Introduction

This hands-on Linux usage lab is performed on the Red Hat Linux 9.x server provisioned and accessible via IBM Technology Zone. Please refer to the [getlinuxserver](#) document to provision your server if you have not already done so.

Lab Environment

The lab exercises will be performed on Red Hat Linux 9.x guests running on IBM Z system Cloud environment hosted from IBM Techzone

Lab Objectives

The objective of this lab is to get familiar with Linux usage by executing some basic Linux commands. The commands are organized into the following categories

1. System information commands
2. User and Group management commands
3. File management commands
4. Networking commands
5. Package management commands
6. Process management commands

Lab Access

1. The lab exercises will be performed on a Red Hat Linux 9.x guest running on IBM Z system Cloud environment hosted by IBM Techzone
2. If you are not connected and if you are using windows workstation, invoke “PuTTY” to connect. For Apple devices use “terminal” utility
 - a. Refer to the [getlinuxserver](#) document to provision your server if you have not already done so
3. We need to be connected as superuser to perform in this lab.

Enter the “**sudo -i**” command as follows.

```
$ sudo -i
```

The following screen is an example where the terminal will show that you are connected to the Linux terminal as “root”

```
root@itzvsi-pvsbiee:~  
login as: UOVTNQM  
Pre-authentication banner message from server:  
  
-----  
Welcome to IBM Technology Zone  
-----  
  
IBM's internal systems must only be used for conducting IBM's business  
or for purposes authorized by IBM management. Use is subject to audit  
at any time by IBM management.  
  
Unauthorized access will be investigated and penalties will be pursued  
in conformance with applicable laws and regulations. If you are not an  
authorized user disconnect now.  
  
End of banner message from server  
Authenticating with public key "imported-openssh-key"  
  
-----  
Welcome to IBM Technology Zone  
-----  
  
IBM's internal systems must only be used for conducting IBM's business  
or for purposes authorized by IBM management. Use is subject to audit  
at any time by IBM management.  
  
Unauthorized access will be investigated and penalties will be pursued  
in conformance with applicable laws and regulations. If you are not an  
authorized user disconnect now.  
  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Register this system with Red Hat Insights: rhc connect  
  
Example:  
# rhc connect --activation-key <key> --organization <org>  
  
The rhc client and Red Hat Insights will enable analytics and additional  
management capabilities on your system.  
View your connected systems at https://console.redhat.com/insights  
  
You can learn more about how to register your system  
using rhc at https://red.ht/registration  
Last login: Thu Oct 16 13:05:11 2025 from 47.188.73.59  
[UOVTNQM@itzvsi-pvsbiee ~]$ sudo -i  
[root@itzvsi-pvsbiee ~]#
```

Now you are connected as a root user.

System information commands

Let's try some of the basic commands to show the information about our Linux system

----- To clear your screen

```
# clear
```

----- To check which user is executing the commands

```
# whoami
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# whoami
root
[root@itzvsi-pvsbiee ~]#
```

----- To get help with any command for usage and command options

```
# whoami --help
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# whoami --help
Usage: whoami [OPTION]...
Print the user name associated with the current effective user ID.
Same as id -un.

    --help      display this help and exit
    --version   output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/whoami>
or available locally via: info '(coreutils) whoami invocation'
[root@itzvsi-pvsbiee ~]#
```

--- To check the hostname

```
# hostname
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# hostname
itzvsi-pvsbiee
[root@itzvsi-pvsbiee ~]#
```

----- To check the date

```
# date
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# date
Thu Oct 16 15:25:00 EDT 2025
[root@itzvsi-pvsbiee ~]#
```

----- You can also concatenate the commands

```
# whoami;hostname;date
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# whoami;hostname;date
root
itzvsi-pvsbiee
Thu Oct 16 15:50:01 EDT 2025
[root@itzvsi-pvsbiee ~]#
```

----- To display the kernel level

```
# uname -a
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# uname -a
Linux itzvsi-pvsbiee 5.14.0-427.42.1.el9_4.s390x #1 SMP Fri Oct 18 14:25:23 EDT 2024 s390x s390x s390x GNU/Linux
[root@itzvsi-pvsbiee ~]#
```

----- To display the architecture

```
# uname -p
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# uname -p
s390x
[root@itzvsi-pvsbiee ~]#
```

----- To check how long the system has been running

```
# uptime
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# uptime
16:04:28 up 1 day, 23:30,  2 users,  load average: 0.15, 0.16, 0.17
[root@itzvsi-pvsbiee ~]#
```

----- To display information about your CPU

```
# lscpu
```

Sample output:

```
Architecture:          s390x
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Big Endian
CPU(s):                 2
  On-line CPU(s) list: 0,1
Vendor ID:             IBM/S390
Machine type:          8562
Thread(s) per core:    1
Core(s) per socket:    1
Socket(s) per book:    1
Book(s) per drawer:    1
Drawer(s):             2
CPU dynamic MHz:       4500
CPU static MHz:        4500
BogoMIPS:              1167.00
Dispatching mode:      horizontal
Flags:                 esan3 zarch stfle msa ldisp eimm dfp edat
                       t sie
Virtualization features:
  Hypervisor:          KVM/Linux
  Hypervisor vendor:    KVM
  Virtualization type: full
Caches (sum of all):
  L1d:                 256 KiB (2 instances)
  L1i:                 256 KiB (2 instances)
  L2d:                 4 MiB (1 instance)
  L2i:                 4 MiB (1 instance)
  L3:                  256 MiB
  L4:                  960 MiB
NUMA:
  NUMA node(s):        1
  NUMA node0 CPU(s):   0,1
Vulnerabilities:
```

----- Another command to display information about your CPU

```
# cat /proc/cpuinfo
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# cat /proc/cpuinfo
vendor_id       : IBM/S390
# processors    : 2
bogomips per cpu: 1167.00
max thread id   : 0
features        : esan3 zarch stfle msa ldisp eimm dfp edat etf3eh highgprs te vx vxd vxe gs vxe2 vxp
sort dflt sie
facilities      : 0 1 2 3 4 6 7 8 9 10 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 30 31 32 33
34 35 36 37 38 40 41 42 43 44 45 47 48 49 50 51 52 53 54 57 58 59 60 61 64 65 69 71 72 73 74 75 76 77
78 80 81 82 129 130 131 133 134 135 138 139 140 146 147 148 150 151 152 155 156 161
cache0          : level=1 type=Data scope=Private size=128K line_size=256 associativity=8
cache1          : level=1 type=Instruction scope=Private size=128K line_size=256 associativity=8
cache2          : level=2 type=Data scope=Private size=4096K line_size=256 associativity=8
cache3          : level=2 type=Instruction scope=Private size=4096K line_size=256 associativity=8
cache4          : level=3 type=Unified scope=Shared size=262144K line_size=256 associativity=32
cache5          : level=4 type=Unified scope=Shared size=983040K line_size=256 associativity=60
processor 0: version = FF,  identification = 05AB48,  machine = 8562
processor 1: version = FF,  identification = 05AB48,  machine = 8562

cpu number      : 0
physical id     : 0
core id        : 0
book id        : 0
drawer id      : 0
dedicated      : 0
address        : 0
siblings       : 1
cpu cores      : 1
version        : FF
identification  : 05AB48
machine        : 8562
cpu MHz dynamic : 4500
cpu MHz static  : 4500

cpu number      : 1
physical id     : 1
core id        : 1
book id        : 1
drawer id      : 1
dedicated      : 0
address        : 1
siblings       : 1
cpu cores      : 1
version        : FF
identification  : 05AB48
machine        : 8562
cpu MHz dynamic : 4500
cpu MHz static  : 4500
[root@itzvsi-pvsbiee ~]# █
```

----- To display information about system memory

```
# cat /proc/meminfo
```


Sample output:

```
[root@itzvsi-pvsbiee ~]# cat /proc/meminfo
MemTotal:       7959032 kB
MemFree:        2377932 kB
MemAvailable:   7226976 kB
Buffers:        4496 kB
Cached:         4706896 kB
SwapCached:      0 kB
Active:         1033196 kB
Inactive:       4005396 kB
Active(anon):    338868 kB
Inactive(anon):  0 kB
Active(file):    694328 kB
Inactive(file):  4005396 kB
Unevictable:     3080 kB
Mlocked:         8 kB
SwapTotal:       0 kB
SwapFree:        0 kB
Dirty:          196 kB
Writeback:       0 kB
AnonPages:       315672 kB
Mapped:          208288 kB
Shmem:           11668 kB
KReclaimable:    271504 kB
Slab:            343068 kB
SReclaimable:    271504 kB
SUnreclaim:      71564 kB
KernelStack:     3696 kB
PageTables:      4172 kB
SecPageTables:    0 kB
NFS_Unstable:    0 kB
Bounce:          0 kB
WritebackTmp:    0 kB
CommitLimit:    3979516 kB
Committed_AS:    1532756 kB
VmallocTotal:   534773760 kB
VmallocUsed:     110472 kB
VmallocChunk:    0 kB
Percpu:          1792 kB
AnonHugePages:   189440 kB
ShmemHugePages:  0 kB
ShmemPmdMapped:  0 kB
FileHugePages:   0 kB
FilePmdMapped:   0 kB
CmaTotal:        0 kB
CmaFree:         0 kB
HugePages_Total: 0
HugePages_Free:  0
HugePages_Rsvd:  0
HugePages_Surp:  0
Hugepagesize:    1024 kB
Hugetlb:         0 kB
DirectMap4k:     8192 kB
DirectMap1M:     4186112 kB
DirectMap2G:     4194304 kB
[root@itzvsi-pvsbiee ~]#
```

----- To check how much free memory is available and swap memory information in mbytes

```
# free -m
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# free -m
              total        used          free      shared  buff/cache   available
Mem:           7772           709          2327         11       4866       7063
Swap:            0            0            0
[root@itzvsi-pvsbiee ~]#
```

----- Another useful command to check how the Linux guest performs is “top”

top

Sample output:

```
top - 18:50:40 up 3 days, 2:16, 2 users, load average: 0.08, 0.13, 0.16
Tasks: 116 total, 1 running, 115 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 7772.5 total, 2324.0 free, 712.6 used, 4866.6 buff/cache
MiB Swap: 0.0 total, 0.0 free, 0.0 used. 7059.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
175169	root	20	0	0	0	0	I	0.3	0.0	0:00.11	kworker/0:3-events
1	root	20	0	107536	16644	10580	S	0.0	0.2	1:20.13	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.50	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp
5	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	slub_flushwq
6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-events_highpri
10	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
12	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_rude_
13	root	20	0	0	0	0	I	0.0	0.0	0:00.00	rcu_tasks_trace
14	root	20	0	0	0	0	S	0.0	0.0	0:02.22	ksoftirqd/0
15	root	20	0	0	0	0	I	0.0	0.0	0:12.08	rcu_sched
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/0
17	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
19	root	rt	0	0	0	0	S	0.0	0.0	0:00.02	migration/1
20	root	20	0	0	0	0	S	0.0	0.0	0:02.06	ksoftirqd/1
22	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H-events_highpri
23	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
24	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	inet_frag_wq
25	root	20	0	0	0	0	S	0.0	0.0	0:00.44	kauditd
27	root	20	0	0	0	0	S	0.0	0.0	0:00.10	khungtaskd
28	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
29	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
30	root	20	0	0	0	0	S	0.0	0.0	0:02.39	kcompactd0
31	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
32	root	39	19	0	0	0	S	0.0	0.0	0:02.40	khugepaged
33	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	cryptd
34	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd
35	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kblockd
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	blkcg_punt_bio
37	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	tpm_dev_wq
38	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md
39	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	md_bitmap
40	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	cio
41	root	-51	0	0	0	0	S	0.0	0.0	0:00.00	watchdogd
42	root	0	-20	0	0	0	I	0.0	0.0	0:04.17	kworker/0:1H-kblockd
43	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
48	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kthrotld
53	root	20	0	0	0	0	S	0.0	0.0	0:05.24	hwrng
54	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kmpath_rdacd
55	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kaluad
56	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mld
57	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ipv6_addrconf
68	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kstrp
69	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kmcheck
74	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/u5:0
187	root	0	-20	0	0	0	I	0.0	0.0	0:04.63	kworker/1:1H-kblockd
365	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rpciod
366	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xprtiod
482	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfstalloc

User and group management commands

Now let us try some typical Linux commands to manage users and groups in Linux.

----- To add a new user “linuxtest” to the system with default options

```
# useradd linuxtest
```

----- To print user and group information for the user “linuxtest”

```
# id linuxtest
```

Sample output:

```
[root@itzvsi-pvsbiee /]# id linuxtest
uid=1001(linuxtest) gid=1002(linuxtest) groups=1002(linuxtest)
[root@itzvsi-pvsbiee /]#
```

----- To add a new group “testers” to the system

```
# groupadd testers
```

----- To add a group to a user use “usermod” command and use -aG option without being removed from the existing groups

```
# usermod -aG testers linuxtest
```

-----Now verify the user “linuxtest” has group “testers” added to it

```
# id linuxtest
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# id linuxtest
uid=1001(linuxtest) gid=1002(linuxtest) groups=1002(linuxtest), 1003(testers)
[root@itzvsi-pvsbiee ~]#
```

----- Now you can switch to the newly created user “linuxtest”

```
# su - linuxtest
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# su - linuxtest
Last login: Thu Oct 16 18:05:26 EDT 2025 on pts/1
[linuxtest@itzvsi-pvsbiee ~]$
```

--- The “/etc/passwd” file contains a list of all users on the system. We can use the “cat” command to view its contents.

```
$ cat /etc/passwd
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:65534:65534:Kernel Overflow User:/:/sbin/nologin
systemd-coredump:x:999:997:systemd Core Dumper:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:998:996:User for polkitd:/:/sbin/nologin
tss:x:59:59:Account used for TPM access:/:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
setroubleshoot:x:997:995:SELinux troubleshoot server:/var/lib/setroubleshoot:/usr/sbin/nologin
sssd:x:996:994:User for sssd:/:/sbin/nologin
cockpit-ws:x:995:993:User for cockpit web service:/nonexisting:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/usr/sbin/nologin
chrony:x:993:990:chrony system user:/var/lib/chrony:/sbin/nologin
tcpdump:x:72:72:/:/sbin/nologin
U0VTNQm:x:1000:1000:ITZ User:/home/U0VTNQm:/bin/bash
systemd-journal-remote:x:992:988:Journal Remote:/var/log/journal/remote:/sbin/nologin
sasauth:x:991:76:Sasauthd user:/run/sasauthd:/sbin/nologin
mongod:x:990:987:mongod:/var/lib/mongo:/bin/false
linuxtest:x:1001:1002:/:/home/linuxtest:/bin/bash
[linuxtest@itzvsi-pvsbiee ~]$
```

----- To remove a user from a specified group you have to be a super user. Exit from current user “linuxtest” and the issue “gpasswd” command with options

```
$ exit
# gpasswd -d linuxtest -g testers
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ exit
logout
[root@itzvsi-pvsbiee ~]# gpasswd -d linuxtest -g testers
Removing user linuxtest from group testers
[root@itzvsi-pvsbiee ~]#
```

----- Now print the user and group information for the user “linuxtest” to see that the “testers” group has been removed

```
# id linuxtest
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# id linuxtest
uid=1001(linuxtest) gid=1002(linuxtest) groups=1002(linuxtest)
[root@itzvsi-pvsbiee ~]#
```

----- The “groupdel” command can be used to remove the “testers” group from the system.

```
# groupdel testers
```

----- Now you can switch back to the user “linuxtest”

```
# su - linuxtest
```

Sample output:

```
[root@itzvsi-pvsbiee ~]# su - linuxtest
Last login: Thu Oct 16 19:16:01 EDT 2025 on pts/1
[linuxtest@itzvsi-pvsbiee ~]$
```

File management commands

Now let us try some of the useful Linux commands to manage the file systems in Linux.

File and Directory operation commands

The following commands will help you navigate, view, and manage files and directories in Linux.

----- To display the absolute path of the current working directory use “pwd” command.

```
$ pwd
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ pwd
/home/linuxtest
[linuxtest@itzvsi-pvsbiee ~]$
```

----- The “cd” command allows you to change your current working directory. Let us change to “tmp” directory under “/”

```
$ cd /tmp
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ cd /tmp
[linuxtest@itzvsi-pvsbiee tmp]$ pwd
/tmp
[linuxtest@itzvsi-pvsbiee tmp]$
```

----- To go directly to your home directory, run “cd ~” command. Then run “pwd” command to display the current working directory

```
$ cd ~
$ pwd
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee tmp]$ cd ~
[linuxtest@itzvsi-pvsbiee ~]$ pwd
/home/linuxtest
[linuxtest@itzvsi-pvsbiee ~]$
```

----- To list the contents of a directory, run “ls” command.

- Use “-a” option to display the hidden files
- Use “-l” option to display detailed information (permissions, owner, size etc.,)
- Use “-R” option to display recursively list everything inside the current directory and its subdirectories
- Use “-t” option to sort display

```
$ ls -aLRt
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ ls -alRt
.:
total 20
-rw-----. 1 linuxtest linuxtest 483 Oct 17 14:19 .bash_history
drwx-----. 3 linuxtest linuxtest 112 Oct 17 14:17 .
-rw-----. 1 linuxtest linuxtest 401 Oct 17 14:17 abc.txt
drwxr-x---. 3 linuxtest linuxtest 18 Oct 16 18:20 .cache
drwxr-xr-x. 4 root      root      38 Oct 16 16:46 ..
-rw-r--r--. 1 linuxtest linuxtest 18 Feb 15 2024 .bash_logout
-rw-r--r--. 1 linuxtest linuxtest 141 Feb 15 2024 .bash_profile
-rw-r--r--. 1 linuxtest linuxtest 492 Feb 15 2024 .bashrc

./.cache:
total 0
drwx-----. 3 linuxtest linuxtest 112 Oct 17 14:17 ..
drwxr-x---. 3 linuxtest linuxtest 18 Oct 16 18:20 .
drwxr-x---. 2 linuxtest linuxtest 22 Oct 16 18:20 rhsm

./.cache/rhsm:
total 0
drwxr-x---. 2 linuxtest linuxtest 22 Oct 16 18:20 .
drwxr-x---. 3 linuxtest linuxtest 18 Oct 16 18:20 ..
-rw-r-----. 1 linuxtest linuxtest 0 Oct 16 18:20 rhsm.log
[linuxtest@itzvsi-pvsbiee ~]$
```

----- To create a new directory use “mkdir” command and then use “cd” command to change our current working directory

```
$ mkdir testdir; cd testdir;
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ mkdir testdir; cd testdir;
[linuxtest@itzvsi-pvsbiee testdir]$
```

----- We can use “cp” command to copy a file from one location to other.

In our case, let us copy a hidden file “/home/linuxtest/.bash_history” to our current directory with a new file name as “abc.txt” and do a “ls” command to display the attributes

```
$ cp /home/linuxtest/.bash_history abc.txt;ls -alF
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ cp /home/linuxtest/.bash_history abc.txt;ls -alF
total 4
drwxr-x---. 2 linuxtest linuxtest 21 Oct 17 14:34 ./
drwx-----. 4 linuxtest linuxtest 127 Oct 17 14:33 ../
-rw-----. 1 linuxtest linuxtest 483 Oct 17 14:34 abc.txt
[linuxtest@itzvsi-pvsbiee testdir]$
```

----- To display the contents of newly copied file “abc.txt” use cat command

```
$ cat abc.txt
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ cat abc.txt
id
touch abc
ls -alF
```

----- To display the specific number of lines from the beginning of the file “abc.txt” use “head” command with “-n” option

```
$ head -n 2 abc.txt
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ head -n 2 abc.txt
id
touch abc
[linuxtest@itzvsi-pvsbiee testdir]$
```

----- To display the last specific number of lines from a file “abc.txt” use “tail” command with “-n” option

```
$ tail -n 2 abc.txt
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ tail -n 2 abc.txt
more abc.txt
exit
[linuxtest@itzvsi-pvsbiee testdir]$
```

----- To check disk space usage on your file systems, use “df” command and the “-h” option will display with human readable format

```
$ df -h
```


Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M   0  4.0M   0% /dev
tmpfs           3.8G   64K  3.8G   1% /dev/shm
tmpfs           1.6G   8.4M  1.6G   1% /run
/dev/vdb2       99G   5.3G   94G   6% /
/dev/vdb1       960M  269M  692M  28% /boot
tmpfs           778M   0  778M   0% /run/user/1000
[linuxtest@itzvsi-pvsbiee testdir]$
```

----- To check how much space is being used by a file or directory on your system use “du” command and the “-h” option will display with human readable format

```
$ du -h /home
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee /]$ du -h /home
du: cannot read directory '/home/U0VTNQM': Permission denied
0      /home/U0VTNQM
0      /home/linuxtest/.cache/rhsm
0      /home/linuxtest/.cache
4.0K   /home/linuxtest/testdir
0      /home/linuxtest/.config/procps
0      /home/linuxtest/.config
24K    /home/linuxtest
24K    /home
[linuxtest@itzvsi-pvsbiee /]$
```

----- You can use additional options to get sortable disk usage for a directory

```
$ du -h /home/linuxtest/ --max-depth=1 | sort -hr
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ du -h /home/linuxtest/ --max-depth=1 | sort -hr
24K    /home/linuxtest/
4.0K   /home/linuxtest/testdir
0      /home/linuxtest/.config
0      /home/linuxtest/.cache
[linuxtest@itzvsi-pvsbiee testdir]$
```

File permission commands

The following commands will help to control who can read, modify, or execute your files in Linux.

Each file in Linux has three sets of permissions:

- u = user (the owner of the file)
- g = group (all users who belong to the same group as the file)
- o = others (everyone else who is not the owner or in the group)

And the types of permissions are:

- r = read (view the contents of a file)
- w = write (modify the contents of a file)
- x = execute (run the file if it's a program or script)

The operators used to change permissions are:

- + = add permission
- - = remove permission
- = = set a permission exactly

----- We can use “ls -l” command to see the permissions

```
$ ls -l
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ ls -l
total 4
-rw-----. 1 linuxtest linuxtest 483 Oct 17 14:34 abc.txt
[linuxtest@itzvsi-pvsbiee testdir]$
```

In the above output the file “abc.txt” you can see the owner as linuxtest and group as linuxtest and the owner can read and write and does not have execute permission.

----- Let us add “execution” permission to that file by using “chmod” command and then use the “ls -l” command to see the new permission levels

```
$ chmod +x abc.txt ; ls -l
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee testdir]$ chmod +x abc.txt ; ls -l
total 4
-rwx--x---. 1 linuxtest linuxtest 483 Oct 17 14:34 abc.txt
[linuxtest@itzvsi-pvsbiee testdir]$
```

Network commands

Now let us try some of the useful network Linux commands

----- We can use “ip address” command to show details about network interfaces. Use “ip help” to find out additional options for the ip command

```
$ ip address ; ip help
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ ip address; ip help
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1450 qdisc fq_codel state UP group default qlen 1000
    link/ether 02:00:23:8a:99:71 brd ff:ff:ff:ff:ff:ff
    altname enc0
    inet 10.241.0.13/24 brd 10.241.0.255 scope global dynamic noprefixroute eth0
        valid_lft 315sec preferred_lft 315sec
    inet6 fe80::23ff:fe8a:9971/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] -batch filename
where  OBJECT := { address | addrlabel | fou | help | ila | ioam | l2tp | link |
                  macsec | maddress | monitor | mptcp | mroute | mrule |
                  neighbor | neighbour | netconf | netns | nexthop | ntable |
                  ntbl | route | rule | sr | stats | tap | tcpmetrics |
                  token | tunnel | tuntap | vrf | xfrm }
      OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
                  -h[uman-readable] | -iec | -j[son] | -p[retty] |
                  -f[amily] { inet | inet6 | mpls | bridge | link } |
                  -4 | -6 | -M | -B | -O |
                  -l[oops] { maximum-addr-flush-attempts } | -echo | -br[ief] |
                  -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
                  -rc[vbuf] [size] | -n[etns] name | -N[umeric] | -a[ll] |
                  -c[olor]}
[linuxtest@itzvsi-pvsbiee ~]$
```

----- The ping command checks if your system can reach another host. “-c “ option tells how many times the ping command is executed

```
$ ping ibm.com -c 5
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ ping ibm.com -c 3
PING ibm.com (184.30.67.31) 56(84) bytes of data.
64 bytes from a184-30-67-31.deploy.static.akamaitechnologies.com (184.30.67.31): icmp_seq=1 ttl=0 time=0.631 ms
64 bytes from a184-30-67-31.deploy.static.akamaitechnologies.com (184.30.67.31): icmp_seq=2 ttl=0 time=0.791 ms
64 bytes from a184-30-67-31.deploy.static.akamaitechnologies.com (184.30.67.31): icmp_seq=3 ttl=0 time=0.776 ms

--- ibm.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.631/0.732/0.791/0.072 ms
[linuxtest@itzvsi-pvsbiee ~]$
```

----- We can use “ss” command to show active network connections, routing tables, and listening ports. Use it with the -tln option to get more information:

- -t shows TCP connections.
- -u shows UDP connections.
- -l shows listening ports.
- -n shows addresses and ports in numeric form.

```
$ ss -tln
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ ss -tln
Netid      State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
udp        UNCONN     0            0            0.0.0.0:68              0.0.0.0:*
tcp        LISTEN     0            128         0.0.0.0:2223            0.0.0.0:*
tcp        LISTEN     0            4096        127.0.0.1:27017         0.0.0.0:*
tcp        LISTEN     0            128         [::]:2223              [::]:*
```

----- We can use “nslookup” command to resolve a domain name to its corresponding IP address.

```
$ nslookup ibm.com
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ nslookup ibm.com
-bash: nslookup: command not found
[linuxtest@itzvsi-pvsbiee ~]$
```

If you receive the above error indicates that the “nslookup utility” is not installed. We will install the needed “bind-utils” package in the next section.

Package management commands

Red Hat-based systems primarily use dnf (or yum in older versions) for package management, which interacts with RPM packages. While rpm is the low-level tool for managing individual packages, dnf and yum provide a higher-level interface for dependency resolution and repository management. Let’s try some of the daily usage package commands as a “root” user. If the active user is “linuxtest” exit to go back to “root” user.

----- Execute the following commands, if you are already a “root” user, no need to exit

```
$ whoami
$ exit
# whoami
```

Sample output:

```
[linuxtest@itzvsi-pvsbiee ~]$ whoami
linuxtest
[linuxtest@itzvsi-pvsbiee ~]$ exit
logout
[root@itzvsi-pvsbiee /]# whoami
root
[root@itzvsi-pvsbiee /]#
```

----- To list all installed packages

```
# dnf list installed
```

Sample output:

```
[root@itzvsi-pvsbiee /]# dnf list installed
Updating Subscription Management repositories.
Installed Packages
NetworkManager.s390x                1:1.52.0-7.el9_6                @rhel-9-for-s390x-baseos-rpms
NetworkManager-libnm.s390x          1:1.52.0-7.el9_6                @rhel-9-for-s390x-baseos-rpms
NetworkManager-team.s390x           1:1.52.0-7.el9_6                @rhel-9-for-s390x-baseos-rpms
NetworkManager-tui.s390x             1:1.52.0-7.el9_6                @rhel-9-for-s390x-baseos-rpms
PackageKit.s390x                     1.2.6-1.el9                     @System
PackageKit-glib.s390x                1.2.6-1.el9                     @System
abattis-cantarell-fonts.noarch       0.301-4.el9                     @System
```

----- To check the status of a specific package

```
# dnf list bind-utils
```

In our case the bind-util is available in the repository, but not installed

Sample output:

```
[root@itzvsi-pvsbiee /]# dnf list bind-utils
Updating Subscription Management repositories.
Last metadata expiration check: 0:06:03 ago on Fri Oct 17 17:56:05 2025.
Available Packages
bind-utils.s390x                32:9.16.23-31.el9_6                rhel-9-for-s390x-appstream-rpms
[root@itzvsi-pvsbiee /]#
```

--- Let's install "bind-utils" package

```
# dnf install -y bind-utils
```

Sample output:

```
[root@itzvsi-pvsbiee /]# dnf install -y bind-utils
Updating Subscription Management repositories.
Last metadata expiration check: 0:08:03 ago on Fri Oct 17 17:56:05 2025.
Dependencies resolved.
=====
Package           Arch      Version              Repository              Size
=====
Installing:
bind-utils        s390x     32:9.16.23-31.el9_6  rhel-9-for-s390x-appstream-rpms  204 k
Installing dependencies:
bind-libs        s390x     32:9.16.23-31.el9_6  rhel-9-for-s390x-appstream-rpms  1.2 M
bind-license      noarch    32:9.16.23-31.el9_6  rhel-9-for-s390x-appstream-rpms   13 k
fstrm             s390x     0.6.1-3.el9          rhel-9-for-s390x-appstream-rpms   29 k
libuv             s390x     1:1.42.0-2.el9_4     rhel-9-for-s390x-appstream-rpms  147 k
protobuf-c        s390x     1.3.3-13.el9         rhel-9-for-s390x-baseos-rpms     38 k
=====
Transaction Summary
=====
Install 6 Packages

Total download size: 1.6 M
Installed size: 4.5 M
Downloading Packages:
(1/6): libuv-1.42.0-2.el9_4.s390x.rpm                29 kB/s | 147 kB    00:05
(2/6): fstrm-0.6.1-3.el9.s390x.rpm                   5.8 kB/s | 29 kB    00:05
(3/6): bind-utils-9.16.23-31.el9_6.s390x.rpm          12 MB/s | 204 kB    00:00
(4/6): bind-libs-9.16.23-31.el9_6.s390x.rpm           239 kB/s | 1.2 MB   00:05
(5/6): bind-license-9.16.23-31.el9_6.noarch.rpm        412 kB/s | 13 kB    00:00
(6/6): protobuf-c-1.3.3-13.el9.s390x.rpm              1.1 MB/s | 38 kB    00:00
-----
Total                                                  323 kB/s | 1.6 MB   00:05
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                                     : 1/1
  Installing : protobuf-c-1.3.3-13.el9.s390x      : 1/6
  Installing : bind-license-32:9.16.23-31.el9_6.noarch : 2/6
  Installing : libuv-1:1.42.0-2.el9_4.s390x       : 3/6
  Installing : fstrm-0.6.1-3.el9.s390x            : 4/6
  Installing : bind-libs-32:9.16.23-31.el9_6.s390x : 5/6
  Installing : bind-utils-32:9.16.23-31.el9_6.s390x : 6/6
Running scriptlet: bind-utils-32:9.16.23-31.el9_6.s390x : 6/6
  Verifying  : fstrm-0.6.1-3.el9.s390x            : 1/6
  Verifying  : libuv-1:1.42.0-2.el9_4.s390x       : 2/6
  Verifying  : bind-libs-32:9.16.23-31.el9_6.s390x : 3/6
  Verifying  : bind-license-32:9.16.23-31.el9_6.noarch : 4/6
  Verifying  : bind-utils-32:9.16.23-31.el9_6.s390x : 5/6
  Verifying  : protobuf-c-1.3.3-13.el9.s390x      : 6/6
Installed products updated.

Installed:
bind-libs-32:9.16.23-31.el9_6.s390x      bind-license-32:9.16.23-31.el9_6.noarch
bind-utils-32:9.16.23-31.el9_6.s390x    fstrm-0.6.1-3.el9.s390x
libuv-1:1.42.0-2.el9_4.s390x            protobuf-c-1.3.3-13.el9.s390x

Complete!
[root@itzvsi-pvsbiee /]#
```

----- Now let us list the package which shows Installed

```
# dnf list bind-utils
```

Sample output:

```
[root@itzvsi-pvsbiee /]# dnf list bind-utils
Updating Subscription Management repositories.
Last metadata expiration check: 0:10:04 ago on Fri Oct 17 17:56:05 2025.
Installed Packages
bind-utils.s390x                32:9.16.23-31.el9_6      @rhel-9-for-s390x-appstream-rpms
[root@itzvsi-pvsbiee /]#
```

----- To uninstall / remove the package

```
# dnf remove -y bind-utils ; dnf list bind-utils
```

Sample output:

```
[root@itzvsi-pvsbiee /]# dnf remove -y bind-utils ; dnf list bind-utils
Updating Subscription Management repositories.
Dependencies resolved.

Package Arch Version Repository Size
-----
Removing:
bind-utils s390x 32:9.16.23-31.el9_6 @rhel-9-for-s390x-appstream-rpms 595 k
Removing unused dependencies:
bind-libs s390x 32:9.16.23-31.el9_6 @rhel-9-for-s390x-appstream-rpms 3.4 M
bind-license noarch 32:9.16.23-31.el9_6 @rhel-9-for-s390x-appstream-rpms 18 k
fstirm s390x 0.6.1-3.el9 @rhel-9-for-s390x-appstream-rpms 47 k
libuv s390x 1:1.42.0-2.el9_4 @rhel-9-for-s390x-appstream-rpms 395 k
protobuf-c s390x 1.3.3-13.el9 @rhel-9-for-s390x-baseos-rpms 61 k

Transaction Summary
-----
Remove 6 Packages

Freed space: 4.5 M
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing : 1/1
  Erasing : bind-utils-32:9.16.23-31.el9_6.s390x 1/6
  Erasing : bind-libs-32:9.16.23-31.el9_6.s390x 2/6
  Erasing : bind-license-32:9.16.23-31.el9_6.noarch 3/6
  Erasing : fstirm-0.6.1-3.el9.s390x 4/6
  Erasing : protobuf-c-1.3.3-13.el9.s390x 5/6
  Erasing : libuv-1:1.42.0-2.el9_4.s390x 6/6
Running scriptlet: libuv-1:1.42.0-2.el9_4.s390x 6/6
  Verifying : bind-libs-32:9.16.23-31.el9_6.s390x 1/6
  Verifying : bind-license-32:9.16.23-31.el9_6.noarch 2/6
  Verifying : bind-utils-32:9.16.23-31.el9_6.s390x 3/6
  Verifying : fstirm-0.6.1-3.el9.s390x 4/6
  Verifying : libuv-1:1.42.0-2.el9_4.s390x 5/6
  Verifying : protobuf-c-1.3.3-13.el9.s390x 6/6
Installed products updated.

Removed:
bind-libs-32:9.16.23-31.el9_6.s390x bind-license-32:9.16.23-31.el9_6.noarch
bind-utils-32:9.16.23-31.el9_6.s390x fstirm-0.6.1-3.el9.s390x
libuv-1:1.42.0-2.el9_4.s390x protobuf-c-1.3.3-13.el9.s390x

Complete!
Updating Subscription Management repositories.
Last metadata expiration check: 0:12:29 ago on Fri Oct 17 17:56:05 2025.
Available Packages
bind-utils.s390x 32:9.16.23-31.el9_6 rhel-9-for-s390x-appstream-rpms
[root@itzvsi-pvsbiee /]#
```

--- Let's install "bind-utils" package again

```
# dnf install -y bind-utils
```

----- Now let us try "nslookup" command to resolve a domain name to its corresponding IP address.

```
# nslookup ibm.com
```

Now the command worked after we installed bind-utils package

Sample output:

```
[root@itzvsi-pvsbiee /]# nslookup ibm.com
Server:          161.26.0.10
Address:         161.26.0.10#53

Non-authoritative answer:
Name:   ibm.com
Address: 184.30.67.31
Name:   ibm.com
Address: 2600:1408:ec00:28b::3831
Name:   ibm.com
Address: 2600:1408:ec00:28a::3831

[root@itzvsi-pvsbiee /]#
```

In this lab we have executed some basic day-to-day Linux commands on a LinuxONE server provisioned from IBM techzone

References

1. "[Red Hat Enterprise Linux](https://www.redhat.com/en)" redhat.com (link resides outside IBM)
2. "<https://samveluibm.github.io/MongoDB-Wildfire-Workshop/>" github.com (link resides outside IBM)

Contact

This hands-on lab guide was created by Sam Amsavelu (samvelu@us.ibm.com) from the IBM Z Washington Systems Center. Please reach out if you have any questions, comments or concerns.