
edureka!
Linux Tutorial

What Will You Learn Today?

edureka!

- ✓ Birth of Linux: Why did Linux become popular?
- ✓ Various distributions of Linux
- ✓ Getting started with Linux
  - ✓ Basic Linux commands
  - ✓ Working with files & directories
  - ✓ Adding/ deleting users
- ✓ Networking in Linux: Using SSH for communication

Why Did Linux Become Popular?

Birth Of Linux

edureka!

1969
→ Birth of **'C'** and **Unix OS**

Birth Of Linux

edureka!

1969
→ Birth of **'C'** and **Unix OS**

1

1970's
→ Growth of Unix because of open-source collaboration
→ Commercial sale of Unix

Birth Of Linux

1969
→ Birth of **'C'** and **Unix OS**

1

1970's
→ Growth of Unix because of open-source collaboration
→ Commercial sale of Unix

2

1980s
→ Companies developing their own Unix:- **IBM**(AIX), **Solaris**(Sun OS), **HP**(HP-UX)....

Mid-to-late 1980s
→ Birth of free software movement → **GNU Project**

Birth Of Linux

edureka!

1969
→ Birth of **'C'** and **Unix OS**

1

1970's
→ Growth of Unix because of open-source collaboration
→ Commercial sale of Unix

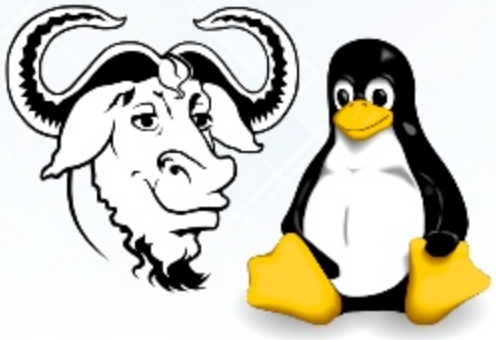
2

1980s
→ Companies developing their own Unix:- **IBM**(AIX), **Solaris**(Sun OS), **HP**(HP-UX)....

Mid-to-late 1980s
→ Birth of free software movement → **GNU Project**

3

1990s
→ Linus Torvalds put the **Linux kernel** source code online.
→ Resulted in usage of **'Linux + GNU'**



Various Linux Distributions

edureka!

01 **RED HAT ENTERPRISE LINUX**

Commercial Linux distribution intended for servers and workstations.

02 **Fedora**

Sponsored by Red Hat, is the foundation for the commercial RHEL

03 **Debian**

Debian is an OS composed only of free, open-source software.

04 **Others**

Other popular distributions include:
Ubuntu, Linux Mint, CentOS, openSUSE / SUSE Linux Enterprise….

Linux's Features

edureka!

Simplified Updates For All Installed S/W

Free Software Licensing

Access To Source Code

Multiple Distributions

Better Malware Protection

Hands-on: Linux Commands

01	02	03	04	05
pwd, clear, ls & cd commands	cat, grep, sort, pipe commands	cp, mv, mkdir, rm, rmdir & user permissions	Linux repository, tar files, env var & regex	Processes, adding users & ssh

Working With Directories: ‘pwd’ & ‘clear’

edureka!

→ Linux provides a **CLI** (Command Line Interface) to communicate with the OS.
→ **CLI** is better for tasks which cannot be performed with the GUI.

Command	Explanation
pwd	Displays the current working directory of the terminal
/	Root directory
echo	Command that writes its arguments to standard output.
su	Used to switch to root user(so that super user permissions can be used to execute commands)
su username	Used to switch to a different user
sudo	Executes only that command with root/ super user privileges
clear	This command is used to clear the terminal screen. Contents will not be deleted but scrolled down

Working With Directories: 'ls' Command

edureka!

$ ls [*options*] [*file* | *Directory*]

ls	This command lists all the contents in the current working directory
ls path	By specifying the path after ls, the content in that path will be displayed
ls –l	Using 'l' flag, lists all the contents along with its owner settings, permissions & time stamp (long format)
ls –a	Using 'a' flag, lists all the hidden contents in the specified directory
ls --author	Using '--author' flag, lists the contents in the specified directory along with its owner
ls –S	Using 'a' flag, sorts and lists all the contents in the specified directory by size
ls *.html	Using '*' flag, lists only the contents in the directory of a particular format
ls –lS > file.txt	Using '>' flag, copies the result of ls command into a text file

Working With Directories: ‘cd’

edureka!

```
$ cd [directory]
```

cd	Changes the directory to the home directory
cd ~	This command also changes the directory to home directory
cd /	Changes the directory to root directory
cd ..	Changes the directory to its parent directory
cd ‘xx yy’	We specify the folder name in inverted commas because there is a space in the folder name

Hands-on: Linux Commands

01	02	03	04	05
pwd, clear, ls & cd commands	cat, grep, sort, pipe commands	cp, mv, mkdir, rm, rmdir & user permissions	Linux repository, tar files, env var & regex	Processes, adding users & ssh

Working With Files: ‘cat’ Command

edureka!

$ cat [*options*] *file1* [*file2..*]

cat	This command is used to display the content of text files and concatenate several files into one
cat -b	This is used to add line numbers to non blank lines
cat -n	This is used to add line numbers to all lines
cat -s	This is used to squeeze blank lines into one line
cat –E	Show $ at the end of line

$ cat > file1.txt

The ‘>’ flag can be used to create a new file and enter text contents from the terminal

$ cat >> file1.txt

The ‘>>’ flag can be used to append text contents to an existing file from the terminal

Working With 'grep' Command

edureka!

We use the 'grep' command to search for a particular string/ word in a text file.

This is similar to "Ctrl+F", but executed via a CLI.

```
$ grep options file1.txt // Returns results for matching string "options"
$ grep –i options file1.txt // Returns the results for case insensitive strings
$ grep –n options file1.txt // Returns the matching strings along with their line number
$ grep –v options file1.txt // Returns the result of lines not matching the search string
$ grep –c options file1.txt // Returns the number of lines in which the results matched search string
```

Working With ‘sort’ Command

edureka!

We use the ‘sort’ command to sort the results of a search either alphabetically or numerically.

Files, file contents and directories can be sorted.

```
$ sort file1.txt // Sorts the contents of file1.txt and returns them in alphabetical order
$ sort File1.txt File2.txt // Sorts the contents of both File1.txt & File2.txt
$ sort –r file1.txt // ‘r’ flag returns the results in reverse order;
$ sort –f file1.txt // ‘f’ flag does case insensitive sorting
$ sort –n file1.txt // ‘n’ flag returns the results as per numerical order
```

Working With '|' Command

edureka!

The '|' command a.k.a 'pipe' command is used to output the result of one command as input to another command.

'|' are used to perform two operations in the same command

```
$ grep dh File1.txt File2.txt | sort // Searches for string 'dh' from both files and sorts the results
$ grep dh File1.txt File2.txt | sort -r // Sorts the results in reverse order
```

Hands-on: Linux Commands

01	02	03	04	05
pwd, clear, ls & cd commands	cat, grep, sort, pipe commands	cp, mv, mkdir, rm, rmdir & user permissions	Linux repository, tar files, env var & regex	Processes, adding users & ssh

Working With Files & Directories: 'cp' Command

edureka!

```
$ cp [options] source destination
```

cp	This command is used to copy files and directories
cp -i	Enters interactive mode; CLI asks before overwriting files
cp -n	Does not overwrite the file
cp -u	Updates the destination file only when source file is different from destination file
cp -R	Recursive copy for copying directories; Copies even hidden files
cp -v	Verbose; Prints informative messages

Working With Files & Directories: 'mv' Command edureka!

$ mv [*options*] *source destination*

mv	This command is used to move files and directories
mv -i	Enters interactive mode; CLI asks before overwriting files
mv -u	Updates the destination file only when source file is different from destination file
mv -v	Verbose; Prints source and destination files

Working With Directories: ‘mkdir’ Command

edureka!

$ mkdir *directory-path*

mkdir	This command is used to create a new directory
mkdir -p	Creates both a new parent directory and a sub-directory
mkdir --parents	This is also used for the same process
mkdir –p file1/{f2,f3,f4}	This is used to create multiple subdirectories inside the new parent directory

Working With Files & Directories: ‘rm’ & ‘rmdir’

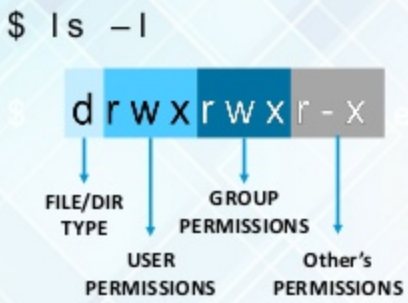
edureka!

```
$ rmdir foldername
$ rm filename/ foldername
```

rmdir	This command is used to remove the specified directory (Empty)
rmdir –p	Removes both the parent and child directory
rmdir –pv	Removes all the parent and sub directories along with the verbose.
rm –r	Removes even non empty directories.
rm –rp	Removes non empty directories including parent and subdirectories.

Working With User Permissions: ‘r’, ‘w’ & ‘x’

edureka!



FILE TYPES

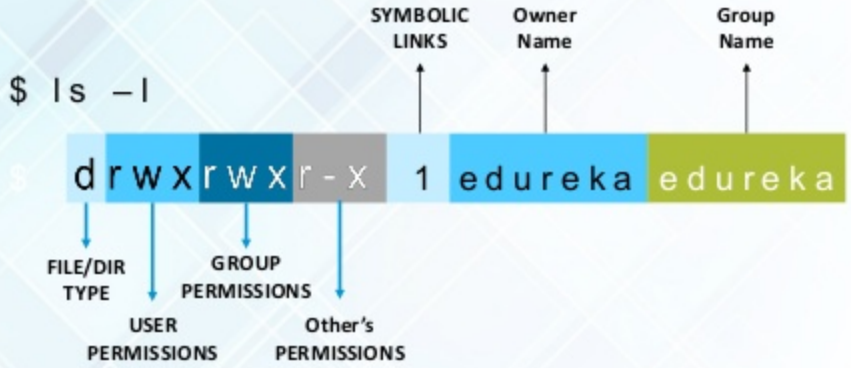
Normal File – ‘-’
Directory – ‘d’
Character Special File – ‘c’
Binary Special File – ‘b’

FILE PERMISSIONS

Read – ‘r’
Write – ‘w’
Execute – ‘x’

Working With User Permissions: ‘r’, ‘w’ & ‘x’

edureka!



FILE TYPES

Normal File – ‘-’
Directory – ‘d’
Character Special File – ‘c’
Binary Special File – ‘b’

FILE PERMISSIONS

Read – ‘r’
Write – ‘w’
Execute – ‘x’

Working With User Permissions: ‘r’, ‘w’ & ‘x’

edureka!

FILE TYPES

Normal File – ‘-’
Directory – ‘d’
Character Special File – ‘c’
Binary Special File – ‘b’

FILE PERMISSIONS

Read – ‘r’
Write – ‘w’
Execute – ‘x’

DENOTIONS

Users – ‘u’
Groups – ‘g’
Others – ‘o’
All – ‘a’

Working With User Permissions: ‘r’, ‘w’ & ‘x’

edureka!

chmod : To change the access permissions of files and directories

chown : To change the owner of files and directories

chgrp : To change the group ownership of file and directories

Commands	Explanation
chmod g+wx filename	This gives the write and execute permission to group members
chmod u=rwx,o-wx filename	This gives the read, write and execute permission to owners, and removes the write and execute ownership from other members
chown username filename	Changes the owner of the specified file
chown username:groupname filename	Changes both the owner and group ownership of the specified file
chgrp groupname filename	Changes the group ownership of the specified file

Hands-on: Linux Commands

01	02	03	04	05
pwd, clear, ls & cd commands	cat, grep, sort, pipe commands	cp, mv, mkdir, rm, rmdir & user permissions	Linux repository, tar files, env var & regex	Processes, adding users & ssh

Working With Linux Repositories

edureka!

Stable versions of most software's will already be available in Linux repositories. Command to install them:

```
$ sudo yum install package-name // For RHEL based systems
$ sudo apt-get install package-name: // For Debian based systems
$ sudo dnf install package-name // For Fedora based systems
```

Java8 Installation

```
$ sudo yum update
$ sudo yum install java-1.8.0-openjdk
$ update-alternatives --config java   // For selecting from various java versions
$ cd
$ sudo gedit .bashrc
# Set the Environment variables here
$ source .bashrc
```

Working With Tar Files

edureka!

When you download a package from the internet, the downloaded file comes in compressed form.

Commands to decompress and compress files in Linux:

gzip: To compress files with .gz format

gunzip: To decompress .gz files

tar: To compress and decompress files with .tar format

```
$ tar –cvf tar-file-name file-name //Compressing files to Tar format
$ tar –xvf tar-file-name //Extracting/ Decompressing files in Tar format
```

Environment Variables

edureka!

Environment variables control the behavior of the software packages installed in Linux.

The path where the packages have been installed will be specified in environment variables.

```
$ printenv // Prints the list of all environment variables
$ echo $HOME // This will print the path of the home directory of the user
$ echo $PATH // It is a colon-separated list of directories in which the shell looks for commands
$ echo $HOSTNAME // This will print the hostname
$ echo $USERNAME // This will print the username
$ echo $LANG // This will print the language being used
$ echo $BASH_VERSION // This will print the version of this instance of bash
```

Regular Expressions

edureka!

Regular Expressions (RegEx) are used to search through data.

It can be piped along with 'grep' command to find patterns of text in the file.

Symbol	Explanation
.	Replaces any character
^	Matches the start of the string
$	Matches the end of the string
*	Matches the preceding character zero or more times
?	Matches the preceding character one or more times
( )	Groups regular expressions
\	Represents special characters

```
$ cat File1.txt | grep ^a
$ grep 'a[0-9]x' File1.txt
$ ls | grep '[xt$]'
```

```
$ ls | grep 'le[0-9].txt‘
$ grep -n ['a*'] File1.txt
$ grep ['d.d'] File1.txt
```

```
$ echo {a..z}
$ echo {4..45}
$ echo 1{a..z}9
```

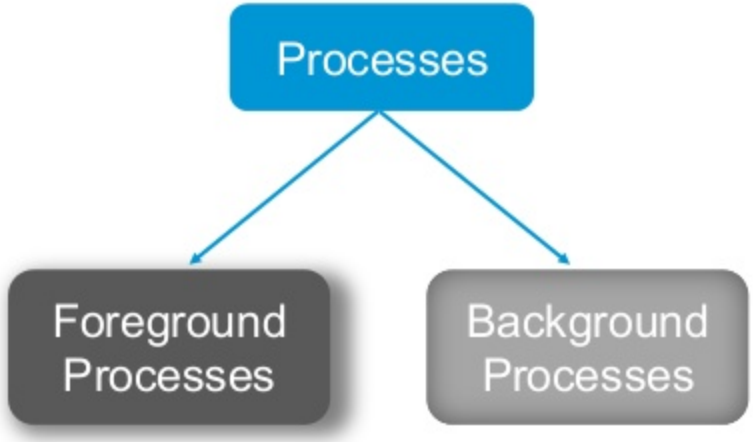
Hands-on: Linux Commands

01	02	03	04	05
pwd, clear, ls & cd commands	cat, grep, sort, pipe commands	cp, mv, mkdir, rm, rmdir & user permissions	Linux repository, tar files, env var & regex	Processes, adding users & ssh

Processes

edureka!

- An instance of a program is called a process.
- Any command given to Linux kernel starts a new process.
- There can be multiple processes (instances) of the same program.




```
$ top
```

PID	USER	PR	NI	VRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
Process ID	User Name	Priority	Niceness Value	Virtual Memory	Physical Memory	Shared Memory	Status	CPU Time	Physical Memory Used	Total CPU Time	Command

Creating Users

edureka!

```
$ sudo useradd user-name // Adding a new user
$ sudo passwd user-name // Setting a password for that user
$ sudo userdel user-name // Deleting that user
```

```
$ sudo groupadd group-name // Adding a new group
$ sudo groupdel group-name // Deleting that group
$ sudo usermod -g group-name user-name // Adding a user to a primary group
```

SSH (Secure Shell) For Remote Machine Access

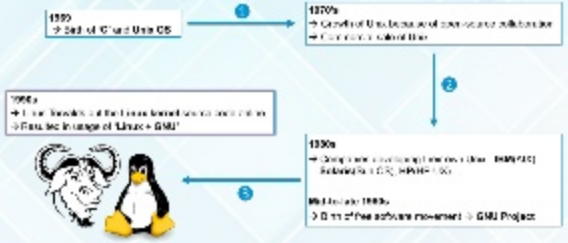
edureka!

```
$ service iptables stop // Drops the IP table
$ sudo chkconfig iptables off // Permanently disbales the IP table

$ sudo gedit /etc/hosts // Add the below IP addresses in both the master and slave's hosts file
master 192.168.56.102
slave 192.168.56.103

$ ip addr show // To show the IP address
$ sudo ip addr del ip-address dev eth1 // Deleting existing IP
$ sudo ip addr add 192.168.56.102/24 dev eth1 // Adding IP at the master's node
$ sudo ip addr add 192.168.56.103/24 dev eth1 // Adding IP at the slave's node

$ ssh master // Running this command at the slave node will give remote access to master
$ ssh slave // Running this command at the master node will give remote access to slave
```

Summary Slide

edureka!

Birth Of Linux



Linux Distributions

Basic Linux Commands

→ Linux provides a CLI (Command Line Interface) to communicate with the OS.
→ CLI is better for tasks which cannot be performed in the GUI.

Command	Explanation
pwd	Displays the current working directory of the terminal
su	Used to switch to root user (as the superuser permissions can be used to execute commands)
su <username>	Used to switch to a different user
sudo	Executes only that command with 'root' super user privileges
clear	The command is used to clear the terminal screen. Contents will not be deleted, but scrolled down

Working With Files & Directories

$ mkdir /directoryname
$ rm -r /directory1/directory2

mkdir	the command is used to create the specified directory [many]
mkdir -p a	to create both the parent and child directory
mkdir -m a	to create the specified directory along with the permissions
mkdir -m a /dir1/{dir2a,dir2b}	To create multiple child directories inside the new parent directory
rm -r	to remove non-empty directories
rm -rf	to remove recursive & forceful including parent and child directories.

Repos, TAR, Env Var, RegEx

Stable versions of most software will already be available in Linux repositories. Command to install them:

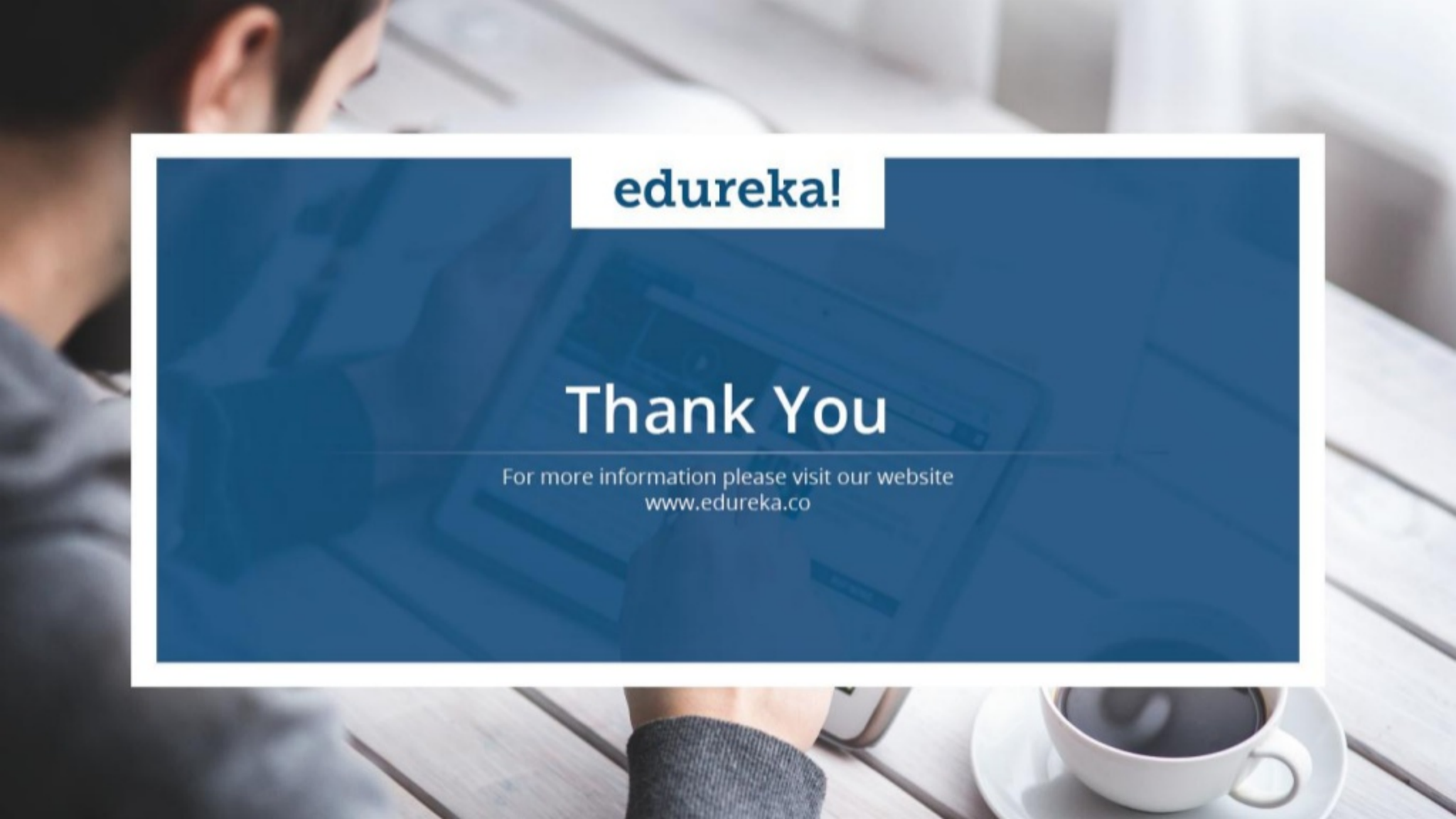
```
$ sudo yum install package-name // For RHEL based systems
$ sudo apt-get install package-name // For Debian based systems
$ sudo dnf install package-name // For Fedora based systems
```

Java8 Installation

```
$ sudo yum update
$ sudo yum install java
$ cd
$ sudo gedit .bashrc
# Set the Environment variables here
$ source .bashrc
```

Processes, Users & SSH

```
$ service iptables stop // Stops the IP table
$ sudo chkconfig iptables off // Permanently disables the IP table

$ sudo gedit /etc/hosts // Add the below IP addresses in both the master and slaves hosts file
master 192.168.56.102
slave 192.168.56.103

$ ip addr show // To view the IP address
$ sudo ip addr del ip-address dev eth1 // Deleting existing IP
$ sudo ip addr add 192.168.56.102/24 dev eth1 // Adding IP in the master's node
$ sudo ip addr add 192.168.56.103/24 dev eth1 // Adding IP in the slave's node

$ ssh master // Running this command at the slave node will give remote access to master
$ ssh slave // Running this command at the master node will give remote access to slave
```


edureka!
Thank You
For more information please visit our website
www.edureka.co