Linux Learnings

Linux Tips and Tricks

Tip 1: How to transfer a file from a host PC to VM PC.

SCP is very useful in transferring file to and from host and remote PC.

Example SCP: scp ./Downloads/docker-desktop-4.28.0-amd64.deb venkatim@9.46.87.181:/home/venkatim/Desktop/Venkat\_Docker\_installable/

Example 2 SCP: ~/.ssh/id\_rsa venkatim@**9.46.93.167:/home/**dp-venkatim-rhel-work/**.ssh/p4\_key**

**Doing scp from remote vm to local mac:**

scp venkatim@9.46.87.181:~/venkat-rel-10-0-1-branch/\_build/x86\_64/datapower/distro-ng/debug\_ag\_vmware.ova ./Desktop/WorkRelated/HandsOn/DataPower/

[Using SCP to transfer files/folder from Local to Remote & from Remote to Local in Linux/Windows/Mac](https://www.youtube.com/watch?v=52NchCNgdw0)

A screenshot of a computer

Description automatically generated

Rsync is another command which does almost the same thing.

Tip 2: Rsync vs SCP when to use each one.

If the number of files to be transferred are less then go for SCP.

For more complicated file transfers use rsync.

Tip 3: How to check if SCP is available in our system:

Try this command: command -v SCP.

**How to find the current OS:**

Uname (To get more details we need to use options)

Uname -v (this command will give the distro name)

Uname -a (this command will give all details about the current distro)

**Memory Related**

=====================

Free -h (This gives the free memory available in system in human readable format)

Free -b/-k/-m/-g (These commands will display free memory in bytes, kilo, mega, giga etc..)

Free -w (this will show buffers and caches separately)

Free -c 2 -s 5 (This command shows free available memory 2 times at interval of 5 secs)

Free -l(Displays the low and high value , which implies the memory utilization)

Free -t(This category displays total mem instead of high and low)

All the memory related values are fetched from the file

$/proc/meminfo

Cat /proc/meminfo

If we need to find a particular type of memory details for ex: cat /proc/meminfo | grep 'Mem'

Will display all memory details of category Mem (Memtotal:, MemFree:, MemAvailable:)

Vmstat (This command displays details about **virtual Memory**)

Vmstat -a(Shows the memory section with active and inactive memory)

For More information see the below link:

[How to check the System Information about Memory usage on Linux ?](https://www.youtube.com/watch?v=W5Nq1L2FJuA)

A screenshot of a computer

Description automatically generated

Command : df. Will give the below details

Filesystem 1K-blocks Used Available Use% Mounted on

/dev/mapper/ubuntu--vg-root 733879300 200294884 533584416 28% /

**How to find the distro Name in Linux**

============================================

cat /etc/\*release\*

**Hot to get details of the port if it is listiening in a dns name**

netstat -an dp-9008-02.rtp.raleigh.ibm.com | grep 9022

Netstat -aunt | grep <portnumber>

p4 revert ... && p4 sync -f ... && rm -rf \_build && make -C toolchain clean && make -C toolchain && jmake -j8 -C third-party world && jmake -j8 -C datapower all-debug

**Which Good text editing software in linux distro**

Gedit <file-name> &

this tool is good for text editing it comes along with line numbers. (Works like note pad)

**How to find ip-address by providing dns name:**

venkatesnm~% nslookup console6.rtp.raleigh.ibm.com

Example:

venkatesnm@Venkatesns-MacBook-Pro ~ % nslookup console6.rtp.raleigh.ibm.com

Server:                192.168.1.254

Address:        192.168.1.254#53

Linux Command Use cases

Use case for **Top** command:

Top command can be used to display snapshots of all the process, It actually displays the overall resource usage of the system. The processes snapshot is captured every 3 seconds once and displayed via top command.

If we are facing resource contention or potential resource contention and if we face slowness in the system then Top command can be useful.

What is CURL:

CURL is an utility that allows us to transfer data to and from a network server using one of the network protocols(http, https, ftp, ftps, sftp, tftp, telnet etc..).

CURL is a multiplatform tool. It works in linux, MAC and Windows.

To get it in MAC use : brew install curl.

To get it in Linux: sudo apt get install curl.

Curl --help

Man curl

How to check TLS handshake:

Curl -v <website address>

What is CURL:

Client URL Request Library.

It helps us to send data over URLs using command line/ CLI. We can upload and download and these are called posts and gets.

Shortcuts in moving to recently run commands:

Use !<number displayed in history>. Ex: !9. ---> this will run a command that is numbers as 9 in the list of history of commands.

To run the last or most recently ran command:

Use the command '!!' this command will run the last ran command.

To search a particular command in the command history and to run that command:

Use the command '!?google' ---> This command will search and find the google string In the command history and directly run the command that is found from the history list.

To list all the command that contains the given string use the command: ~$history | grep ping

This will list all the command that has the string ping.

What is SCP:

SCP is a utility used to transfer files from host to remote or remote to host.

The format of the command is as below:

scp <root\_system\_file\_path> root@<dest\_system\_ip>:<dest\_system\_store\_loc\_path>

 scp .ssh/id\_rsa venkatim@9.46.87.181:/home/venkatim/.ssh/id\_rsa

How to list the installed components of a particular type:

apt list --installed | grep docker

[How To Use The scp Command to Copy a File From Remote to Local (and vice versa)](https://www.youtube.com/watch?v=q2OHvlr081s&t=225s)

A computer screen with arrows pointing to a computer screen

Description automatically generated

[Transferring files with the scp Command (Linux Crash Course Series](https://www.youtube.com/watch?v=Aa7tKMmeFZI))

A person pointing at a computer screen

Description automatically generated

**How to search a pattern present in a file**

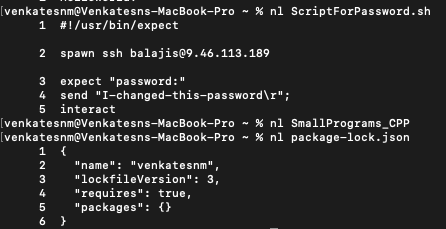
grep -inr "Text" folder/to/be/searched/

grep -Ril "text-to-find-here" /

Good link: <https://www.cyberciti.biz/faq/howto-search-find-file-for-text-string/#google_vignette>

**To display contents of a file(directly in the same bash screen) with line number:**

nl <file\_name>



ps -eF | grep p4

**How to log the build progress**

| tee -a file2.txt

The below grep command will search for 'CreateZip, ket or ExtractZip, ket' the result also displays previous and next line.

grep -Ei -b1 '(create|extract)zip.\*, ket' datapower/router/drWebGUI.cpp

Ex:

$grep -Ei -b1 '(create|extract)zip.\*, ket' datapower/router/drWebGUI.cpp

64787- MAKE\_QNAME(uri, qp->GetLocalCode("create-zip", Qref::FOREVER())),

64865: CreateZip, ketString, kSideEffects);

64914- f->AddUserDataArgument(webguikey, offset\_of(ZipArgs, mProcessor));

--

65393- MAKE\_QNAME(uri, qp->GetLocalCode("extract-zip", Qref::FOREVER())),

65472: ExtractZip, ketString, kSideEffects);

65522- f->AddUserDataArgument(webguikey, offset\_of(ZipArgs, mProcessor));

Code navigation in VIM

For Code Navigation use gedit.

<https://www.geeksforgeeks.org/gedit-command-in-linux/>

Code Navigation using Emacs:

[The Basics of Emacs as a Text Editor](https://www.youtube.com/watch?v=jPkIaqSh3cA)

A computer screen with text overlay

Description automatically generated

Navigation shortcuts in Emacs:

To zoom in the file: ctrl + x + ctrl + '+'

To zoom out in the file: ctrl + x + ctrl + '-'

To Open a File: ctrl + x + ctrl + f (This opens the file search edit box) (you can go to root path to use like folder navigator)

To move forward use: ctrl + f

To move backward use: ctrl + b

To move next item use: ctrl + n (to move to next line)

To move to previous item ctrl + p (to move to previous line)

To close a file ctrl + x, ctrl + c

To move to previous line use: ctrl + p

To move forward by word by word use: command + f

To move backward by word by word use: command + b

To move to the end of the line use: ctrl + e

To move beginning of the line use: ctrl + a

For horizontal split use: ctrl + x + 2

To open different file in horizontal split use: ctrl + x + 2f

To close file opened in horizontal split use: ctrl x + 0 + c

For vertical split use: ctrl + x + 3

To open different file in vertical split use: ctrl + x + 4f

How to switch cursor between split windows: ctrl + x + o

How to scroll:

* Ctrl + b: Moves the page up
* Ctrl + f: Moves the page down
* Ctrl + u: Moves the page up half a page
* Ctrl + d: Moves the page down half a page
* j: Moves the cursor down one line
* k: Moves the cursor up one line
* 2j: Moves the cursor down two lines
* 2k: Moves the cursor up two lines
* zt: Moves the screen so the cursor is on the top (vertical)
* zb: Moves the screen so the cursor is on the bottom (vertical)
* zz: Moves the screen so the cursor is at the center (vertical)
* zs: Moves the screen so the cursor is at the start (horizontal
* To go to the beginning of the file :1
* To go to the end of the file press shift + g

**1. Basic Editing:**

* **C-x C-f**: Open a file (find file).
* **C-x C-s**: Save the current file.
* **C-x C-c**: Exit Emacs.
* **C-g**: Cancel a command.
* **C-/** or **C-x u**: Undo the last change.
* **C-space**: Start selecting a region (mark a position).
* **C-w**: Cut (kill) the selected region.
* **M-w**: Copy the selected region.
* **C-y**: Paste (yank) the most recently cut/copied text.
* **C-k**: Cut (kill) from the cursor to the end of the line.
* **C-e**: Move to the end of the line.
* **C-a**: Move to the beginning of the line.
* **C-s**: Search forward.
* **C-r**: Search backward.

**2. Movement:**

* **C-n**: Move to the next line.
* **C-p**: Move to the previous line.
* **C-f**: Move forward one character.
* **C-b**: Move backward one character.
* **M-f**: Move forward one word.
* **M-b**: Move backward one word.
* **C-v**: Scroll down one screen.
* **M-v**: Scroll up one screen.
* **M-<**: Move to the beginning of the buffer.
* **M->**: Move to the end of the buffer.

**3. Window and Buffer Management:**

* **C-x 2**: Split the window horizontally.
* **C-x 3**: Split the window vertically.
* **C-x 0**: Close the current window.
* **C-x 1**: Close all other windows (focus on the current one).
* **C-x o**: Switch between open windows.
* **C-x b**: Switch to another buffer.
* **C-x k**: Kill (close) the current buffer.

**4. Help:**

* **C-h t**: Open the Emacs tutorial.
* **C-h k**: Describe a keybinding (shows what a key does).
* **C-h f**: Describe a function.
* **C-h v**: Describe a variable.
* **C-h a**: Search for functions and commands by keyword.

**5. Macros:**

* **C-x (**: Start recording a macro.
* **C-x )**: Stop recording the macro.
* **C-x e**: Play (execute) the last macro.

**6. File and Directory Management:**

* **C-x d**: Open dired (directory browser mode).
* **C-x C-w**: Save the current buffer with a new name (write file).
* **C-x i**: Insert a file at the cursor position.

**7. Text Manipulation:**

* **M-u**: Convert the next word to uppercase.
* **M-l**: Convert the next word to lowercase.
* **M-c**: Capitalize the next word.
* **M-d**: Delete the next word.
* **M-backspace**: Delete the previous word.

**8. Command Execution:**

* **M-x**: Execute an Emacs command by name. (This is one of the most powerful commands in Emacs.)

**9. Repetitions:**

* **C-u**: Prefix command to repeat actions (e.g., C-u 5 C-n moves down 5 lines).

**Key Legend:**

* **C-**: Hold the Control key (e.g., C-x means "hold Control and press x").
* **M-**: Hold the Meta key (usually Alt or Esc on modern keyboards).

Pattern Matching, File searching tips:

<https://www.geeksforgeeks.org/grep-command-in-unixlinux/>

1. To Display count of Number of matches

Ex: grep -c "pattern-to-search" fileName.txt

2. To Display file names that matches the pattern.

Ex: grep -l "pattern-to-search" \* Or Ex: grep -l "pattern" filename1.txt filename2.txt filenameN.txt

3. To search for the whole word in a file:

Ex: grep -w "pattern" fileName.txt

4. Displaying only the matched pattern

By default grep displays the entire line. We can make grep to display only the matched string by using -o.

Ex: grep -o "pattern-to-search" fileName.txt

5. To display line numbers when displaying output of grep.

Ex: grep -n "pattern" fileName.txt

6. To Invert pattern match using grep.

Ex: grep -v "pattern" filename.txt.

7. To find Matching lines that start with a string

Ex: grep "^pattern" fileName.txt.

8. To find matching lines that end with a string

Ex: grep "pattern$" filename.txt

Setting up SSH connection permanently to VSCode

<https://code.visualstudio.com/docs/remote/troubleshooting>

Details copied from .ssh/config

Host 9.46.87.181

  HostName 9.46.87.181

  User venkatim

Host dp-venkatim-work

  HostName dp-venkatim-work

  User venkatim

Run this below script in the command line

Ensure that the ssh is already created

Also visit the .ssh/config file and get the details from the file

$USER\_AT\_HOST="venkatim@9.46.87.181"

$PUBKEYPATH="$HOME\.ssh\id\_rsa.pub"

$pubKey=(Get-Content "$PUBKEYPATH" | Out-String); ssh "$USER\_AT\_HOST" "mkdir -p ~/.ssh && chmod 700 ~/.ssh && echo '${pubKey}' >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"

If I Run the above script then we'll get password request prompt and that will be the only time we may have to enter the password and from then on all the time the VSCode will open automatically.

Check more details here:

<https://code.visualstudio.com/docs/remote/troubleshooting>

[SSH using VSCode to connect to VM or Workstation](https://www.youtube.com/watch?v=ar_ZjFu0FP4)

A screenshot of a computer

Description automatically generated

<https://www.youtube.com/watch?v=0Bjx3Ra8PRM>

[SSH into Remote VM with VS Code | Tunneling into any cloud | GCP Demo](https://www.youtube.com/watch?v=0Bjx3Ra8PRM)

A blue and white background with text and symbols

Description automatically generated

MAC productivity

MAC access tips:

Screenshot commands:

How to take screenshot (full screen) = command + Shift + 3

How to take screenshot (selection region) = Command + shift + 4

Cut and Paste command:

How to move a file or folder from one location to another.

Press Command + C and then press Command + Option + V

How to search a file or find a file based on its content:

Use the Spotlight search --> Command + space

When searching if we want to filter the result then add the file name and mention Kind:<file extension> for filtering.

From the result we can peek at the file by pressing space bar.

To move to its file location press command + enter.

TMUX setup

[you need to learn tmux RIGHT NOW!!](https://www.youtube.com/watch?v=nTqu6w2wc68)



[Tmux will SKYROCKET your productivity - here’s how](https://www.youtube.com/watch?v=niuOc02Rvrc)



[What is TMUX?](https://www.youtube.com/watch?v=9AECD1p8wVU)



[Complete tmux Tutorial](https://www.youtube.com/watch?v=Yl7NFenTgIo)



[Part 1 - What is TMUX](https://www.youtube.com/watch?v=LDGMF3HPKe8)

A red square with white text

Description automatically generated

Tmux offers a variety of shortcut keys for efficient session management, pane splitting, window creation, and more. Here’s a list of the most commonly used tmux shortcuts.

**General Tmux Shortcuts**

By default, tmux shortcuts are accessed by pressing **Ctrl + b** (often referred to as the tmux prefix) followed by another key. You can change the prefix to another key if desired, but **Ctrl + b** is the default.

**Pane Management**

* **%**: Split the current pane vertically.
* **"**: Split the current pane horizontally.
* **Arrow keys** (or **h/j/k/l** with Vim key bindings): Move between panes.
* **x**: Close the current pane.
* **z**: Toggle the pane between full-screen and normal size.
* **!**: Break the current pane into a new window.

**Window Management**

* **c**: Create a new window.
* **,&**: Rename the current window.
* **,**: Rename the current window.
* **p**: Move to the previous window.
* **n**: Move to the next window.
* **&**: Kill the current window.
* **0 - 9**: Switch to window number 0 through 9.

**Session Management**

* **d**: Detach from the current session.
* **Ctrl + b + s**: List all sessions.
* **:kill-session**: Kill the current session (type after pressing the prefix key).

**Resizing Panes**

* **:resize-pane -U**: Resize pane up.
* **:resize-pane -D**: Resize pane down.
* **:resize-pane -L**: Resize pane left.
* **:resize-pane -R**: Resize pane right.

Or you can use:

* **Ctrl + b + Alt + Arrow Keys**: Resize the current pane in the direction of the arrow key.

**Copy Mode (Scrollback)**

* **[**: Enter copy mode.
* **Esc**: Exit copy mode.
* **Space**: Start selection in copy mode.
* **Enter**: Copy selected text.
* **q**: Exit copy mode.
* **Ctrl + u**: Page up.
* **Ctrl + d**: Page down.

**Miscellaneous Shortcuts**

* **?**: Show all key bindings.
* **:**: Enter command mode to type tmux commands manually.
* **t**: Display the time and date in the status line.
* **Ctrl + b + r**: Reload the tmux configuration file (~/.tmux.conf).

These are some of the main shortcuts available in tmux, though additional key bindings can be configured in the .tmux.conf file. Adjusting tmux shortcuts for personal preference is common, so you might encounter different key bindings based on customized configurations.

**How to connect to list all previous sessions of tmux**

$tmux ls

**How to connect to a particular session listed by tmux ls**

$tmux a -t <index-Num/Session-Name>

Ex:

$ tmux a -t 1

or

$ tmux a -t Name-Of-Session

**How to kill a session**

Tmux kill-session -t <session-name/ session-index>

Ex:

$tmux kill-session 0

$tmux kill-session <session-name>

**How to create a new session**

$tmux new -s <session-name>

Learning Regular Expression

sed -i -e "s/^<product>IDG:/<product>IDG:venkat:/" dev-xu-368442.scrypt4

The above command will replace the pattern <product>IDG: -----> to <product>IDG:venkat

<https://www.geeksforgeeks.org/how-to-read-specific-lines-from-a-file-in-python/>

sed -i -e "s/^<BillDetail>IDG:/<BillDetail>BILL:1234:venkat:/" <file-name>

**The below expression will be useful to change the following string from "<product>IDG:9005:</product>" to "<product>IDG:9005:Tenant</product>"**

Ex 1: sed -i -e "s/:<\/product>END$/:Tenant<\/product>FINAL/" config.scrypt4

Ex 2: sed -i -e 's/:<\/product>/:Tenant<\/product>/' config.scrypt4

**This below expression is replacing the given content**

sed -i "10s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>${new\_value}<\/BillOffset>/" "$file\_name"

Ex: sed -i "10s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>$4293<\/BillOffset>/" "config.xml"

Below line is working : It is used to replace the content of <BillOffset>1234</BillOffset> wit the new value given

sed -i "10s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>${new\_value}<\/BillOffset>/" "$file\_name"

Ex: sed -i "10s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>4264<\/BillOffset>/" config.xml

**Learning Regular expressiion and SED usage:**

Prompt give: given the below line :</BillDetail>. Suggest a regular expression to change the string as following :Rajeev</BillDetail>

sed -i -e 's/:<\/product>/:Tenant<\/product>/' config.xml

Explanation:

To modify the string </BillDetail> and replace it with :Rajeev</BillDetail>, you can use the following regular expression:

**Regular Expression:**

s/<\/BillDetail>/:Rajeev<\/BillDetail>/

**Explanation:**

* **<\/BillDetail>**: Matches the closing </BillDetail> tag.
* **:Rajeev<\/BillDetail>**: The replacement text where :Rajeev is added before the </BillDetail> tag.

**sed Command Example:**

You can use this regular expression with sed as follows:

sed -i -e 's/<\/BillDetail>/:Rajeev<\/BillDetail>/' <file-name>

**What this does:**

* It searches for the </BillDetail> tag.
* It replaces it with :Rajeev</BillDetail>, appending :Rajeev before the closing tag.

**Learning Regular expression Example 1**

**Prompt Given**: Using regular expression in shell script, how to replace a integer value present between <BillOffset>xxxx</BillOffset> tag. This <BillOffset> tag is present at 10th line inside the file.

GPT Returned:

To replace the integer value between <BillOffset> and </BillOffset> in a specific line (e.g., line 10) of a file using a **shell script** with regular expressions, you can combine sed with a command to extract the 10th line and modify its value.

Here's how you can use a shell script with sed to perform this:

Shell Script:

#!/bin/bash

# Define the file name

file\_name="your\_file.xml"

# Define the line number where <BillOffset> is located (e.g., line 10)

line\_number=10

# Use sed to capture the number between <BillOffset> tags, add 8, and replace it

sed -i "${line\_number}s/\(<BillOffset>\)\([0-9]\+\)\(<\/BillOffset>\)/echo \1$((\2 + 8))\3/e" "$file\_name"

echo "BillOffset value updated successfully in line $line\_number!"

**Explanation:**

1. **${line\_number}s/.../.../**:
   * **This restricts the operation to the 10th line (line\_number=10), using the line number in the sed command.**
2. **(<BillOffset>)([0-9]+)(<\/BillOffset>)**:
   * **(<BillOffset>): Captures the opening <BillOffset> tag.**
   * **([0-9]+): Captures the integer value between the tags (one or more digits).**
   * **(<\/BillOffset>): Captures the closing </BillOffset> tag.**
3. **echo \1$((\2 + 8))\3/e**:
   * **\1: Refers to the first captured group, <BillOffset>.**
   * **\2: Refers to the second captured group (the integer value). $(($2 + 8)) adds 8 to this value.**
   * **\3: Refers to the third captured group, </BillOffset>.**
   * **/e: Executes the command (in this case, it replaces the old integer value with the new value plus 8).**

When tried to execute the above, Got the Error as follows: getting error in this above code update\_bill\_offset.sh : \2 + 6: syntax error: operand expected (error token is "\2 + 6")

**Now gave this as a prompt:**

The error occurs because shell arithmetic in sed does not directly support regular expression group captures like \2. The sed command's /e flag allows for external commands, but you cannot use \2 for arithmetic in this way.

We can fix this by ensuring the captured value is passed to bash for arithmetic processing in a more robust way. Here's an updated solution that works by extracting the value, performing the arithmetic outside of sed, and reinserting the result.

**Updated Shell Script**

#!/bin/bash

# Define the file name

file\_name="your\_file.xml"

# Define the line number where <BillOffset> is located (e.g., line 10)

line\_number=10

# Use sed to extract the number from <BillOffset> tags, calculate the new value, and replace it

sed -i "${line\_number}s/\(<BillOffset>\)\([0-9]\+\)\(<\/BillOffset>\)/\1$((\2 + 8))\3/" "$file\_name"

echo "BillOffset value updated successfully in line $line\_number!"

**Explanation of Fix:**

1. **\([0-9]\+\)**: This matches the integer value between the <BillOffset> tags and captures it in \2.
2. **$((\2 + 8))**: Instead of trying to use \2 directly inside the arithmetic expression, this updated command correctly uses the $(( )) syntax for shell arithmetic within sed.
   * **This will add 8 to the captured value.**

**Key Fix:**

* The **previous mistake** was using the /e flag to try to process arithmetic directly in the regex replacement, which doesn't work well with shell captures.
* Now, \2 is correctly evaluated inside $((\2 + 8)).

**Example:**

For a file with the 10th line as:

Result: Got error again

Error: : \2 + 6: syntax error: operand expected (error token is "\2 + 6")

Provided the same as prompt again.

The error you're encountering (\2 + 6: syntax error: operand expected) indicates that sed is not interpreting the capture group (\2) as a valid arithmetic operand. Unfortunately, sed does not support arithmetic operations directly on captured groups in this way.

To overcome this, you need to extract the value, perform arithmetic in the shell, and then replace the value back into the file. Here’s how you can do it using a combination of sed, shell variable manipulation, and a while loop:

**Corrected Shell Script**

#!/bin/bash

# Define the file name

file\_name="your\_file.xml"

# Define the line number where <BillOffset> is located (e.g., line 10)

line\_number=10

# Extract the value of BillOffset from the specific line

bill\_offset=$(sed -n "${line\_number}s/.\*<BillOffset>\([0-9]\+\)<\/BillOffset>.\*/\1/p" "$file\_name")

# Check if extraction was successful

if [ -z "$bill\_offset" ]; then

echo "Could not find a BillOffset value on line $line\_number."

exit 1

fi

# Add 8 to the extracted BillOffset value

new\_bill\_offset=$((bill\_offset + 8))

# Replace the old BillOffset value with the new value

sed -i "${line\_number}s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>${new\_bill\_offset}<\/BillOffset>/" "$file\_name"

echo "BillOffset value updated successfully to $new\_bill\_offset on line $line\_number!"

**Explanation:**

1. **Extracting the BillOffset value**:
   * **sed -n "${line\_number}s/.\*<BillOffset>\([0-9]\+\)<\/BillOffset>.\*/\1/p":**
     + **This sed command reads the specific line (in this case, line 10) and captures the integer value between <BillOffset> and </BillOffset>.**
     + **\([0-9]\+\) captures the digits between the tags, and \1 is used to output the captured value.**
     + **The -n flag ensures that sed prints only the value and nothing else.**
2. **Check for extraction success**:
   * **If the bill\_offset variable is empty (meaning it couldn't find a <BillOffset> tag), the script exits with an error message.**
3. **Performing arithmetic**:
   * **The arithmetic operation is performed using new\_bill\_offset=$((bill\_offset + 8)).**
4. **Replacing the old value**:
   * **sed -i "${line\_number}s/<BillOffset>[0-9]\+<\/BillOffset>/<BillOffset>${new\_bill\_offset}<\/BillOffset>/":**
     + **This sed command replaces the old value with the new value inside the <BillOffset> tags on the specific line.**
     + **[0-9]\+ matches any sequence of digits.**

OVA file (ILMT Sub section)

What is an OVA? The below description is from IBM genAI.

An OVA (Open Virtualization Appliance) file is a compressed file format used to package and distribute virtual appliances, such as virtual machines (VMs), on a network. It is commonly used to distribute virtual appliances on public clouds, such as Amazon Web Services (AWS) or Microsoft Azure.

The OVA file format is a standard format used by various virtualization platforms, such as VMware ESXi, Oracle VirtualBox, and Citrix XenServer. An OVA file contains all the files and configuration settings required to create a virtual appliance, including the operating system, applications, and data.

When you download an OVA file from a website or from an email attachment, it will typically have a file extension of .ova, which stands for "open virtualization appliance". To use the OVA file, you will need to import it into your virtualization platform, such as VMware ESXi, Oracle VirtualBox, or Citrix XenServer.

Once imported, you can create multiple instances of the virtual appliance, each with its own unique configuration and data. For example, you could create a development environment using an OVA file, and then create a separate production environment using the same OVA file.

OVA files are particularly useful for deploying complex applications and environments quickly and efficiently, as they allow you to easily replicate environments and reduce the risk of errors associated with manual setup. They are also a popular choice for deploying virtual appliances on public clouds, where they provide a convenient way to scale up or down resources as needed.

What is an OVA file?

An ova file is a virtual appliance used in a virtualization applications such as VMware workstation and oracle VM Virtual box and Nutanix.

It is a Package that contains files that describes a virtual machine, specially customized operating system. which includes an OVF descriptor files, Optional manifest (.MF) and certificate files, and other related files in tar format.

**How to access ESXi Vmware**

=================================

At least for now, the login issues seem resolved.

The link to the ESXI is  <https://dpx3650c.dp.rtp.raleigh.ibm.com/Login> credentials will be isldev:jan0111j.

Your going to want to click “Create/Register VM”Then, “Deploy a virtual machine from an OVF or OVA file”Upload the OVA file you built on your workstation.

Give the virtual machine a name that indicates who owns it and what it’s for.

In this case I’d go with something like “Bhiutras-ILMT”Storage should just be the one option.Accept the llicense agreement.Keep deployment options default.

Don't need additional settings.

Click Finish.It'll take some time for ESXI to provision the VM, you'll see it as a task at the bottom of the console.

Once it comes up you'll have access to the cli through the console view, and those ilmt commands should work now.

**How to setup IP in ESXi VM:**

* Login as admin; co ; ethernet eth0 ;
* ip-address 9.42.102.152/24
* ipv4-default-gateway 9.42.102.1 ; show ; exit
* To enable web management in OVA VM from CLI -
* Login as admin; co; show web-mgmt; web-mgmt ; admin-state enabled ; exit
* Enable ssh access from outside -  ssh 0.0.0.0
* At the end write memory ; co ; write mem ; y ; exit

Setting up screens in Linux

How to setup screen in linux

**To get create a new screen**

Ex: screen -S DP\_access\_Tenat\_LandLord

**To list the Number of screens**

Ex: screen -ls

**To get into a particular screen that was created**

Ex: screen -x DP\_access\_Tenat\_LandLord

**If there are multiple screens displaying and if we want to close or remove a screen from the list of screens execute the below command**

Ex: screen -d -r 1228131.1223505.DP\_access\_Tenat\_LandLord

And then enter

Ex: CTRL+a CTRL+k

<https://www.geeksforgeeks.org/screen-command-in-linux-with-examples/>