



Conversation with:
Skyla Easlie
Linux Systems Administrator

Systems Administration Adv. Challenge Chat Transcript

Skyla sent:

Hello there, @playerone! You must be my new intern! 😊
I'm Skyla Easlie, I'm a Linux Systems Administrator here at TryCyber Credit Union.

Skyla sent:

I'm going to be showing you how to perform some core **Systems Administration** tasks. I'll be doing that by having you help me accomplish some of the simpler tasks I take care of using the **Ubuntu 22.04 Linux** server you are looking at now. ✨

Skyla sent:

The tasks we will be doing today will be completed using the **Terminal**. It can be intimidating if you have not used it before. So let me know if you want a brief tutorial on it.

Skyla sent:

Are you ready to get started? 😊

Participant sent:
Terminal tutorial, please!

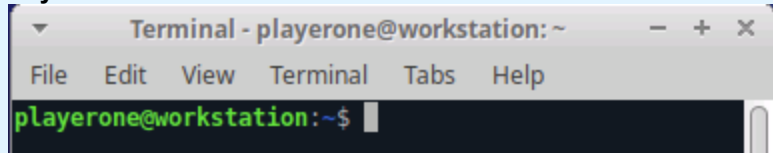
Skyla sent:

The **Terminal** is primarily used to access an application called a shell. We interact with shells via a **command-line interface (CLI)** where we input and execute text-based commands on a system.

Skyla sent:

Let's open the Terminal on this system now and run a few basic commands so you get the hang of it. You can open the **Terminal** by double-clicking the icon on the desktop named **Terminal Emulator**. It should look like this once you have it open...

Skyla sent:



Participant sent:
I've opened the Terminal.

Skyla sent:

Great. Basically, you type commands into the shell within the Terminal and then hit **Enter** or **Return** on the keyboard to run the command.

Skyla sent:

To properly utilize any CLI command, you must understand its format, options, and other arguments. The **format** defines the structure and order of options and other arguments, the **options** define or modify behavior, and the **arguments** are command defined input types (e.g., files, directories, users, software packages, etc.).

Skyla sent:

Additionally, all CLI commands on this system are case-sensitive, so pay close attention to the capitalization of letters.

Skyla sent:

The basic command format is...

```
command [OPTIONS] [ARGUMENTS]
```

However, it varies a lot from command to command, and options and other arguments are not always required.

Skyla sent:

An example of a command that does not require any options or other arguments is `whoami`. If you type `whoami` into the shell and then hit `Enter` or `Return` on the keyboard, the `whoami` command will output the name of the user running the command into the Terminal. It should look like this if you run that command...

Skyla sent:

```
playerone@workstation:~$ whoami  
playerone
```

Participant sent:
What's a more involved example?

Skyla sent:

I'll give you a more complex example using the `ls` command.

Skyla sent:

In this example, we will use the `ls` command to list detailed information about the entire contents of the `Templates` directory (i.e., folder) in your (playerone's) `home` directory. Our example `ls` command will use the following format...

```
ls [OPTIONS] [DIRECTORY]
```

Note that in this case, the command's argument requires the input to be a directory.

Skyla sent:

The actual command we want to run is...

```
ls -a -l /home/playerone/Templates
```

Skyla sent:

The `-a` and the `-l` are both options that modify the `ls` command's behavior, and `/home/playerone/Templates` is our argument which is a directory provided in the form of a path.

Skyla sent:

For this command's argument, we must provide the **path** to the directory from the root of the file system; otherwise, the system won't know which directory named `Templates` we are referring to.

Skyla sent:

And finally, when you run that command, the output should look like this...

Skyla sent:

```
playerone@workstation:~$ ls -a -l /home/playerone/Templates/
total 24
drwxr-xr-x  2 playerone playerone 4096 Jul 12 21:36 .
drwxr-x--- 14 playerone playerone 4096 Jul 12 21:36 ..
-rw-r--r--  1 playerone playerone 6859 Apr  9 2022 'OpenDocument Spreadsheet.ods'
-rw-r--r--  1 playerone playerone 7388 Apr  9 2022 'OpenDocument Text.odt'
-rw-r--r--  1 playerone playerone   0 Apr  9 2022 'Plain Text.txt'
```

Participant sent:
Got it. Any extra notes?

Skyla sent:

The Terminal and shells are incredibly powerful and versatile tools. Not all commands and programs follow the general structure I've provided you with here. Unfortunately, we only have time to cover the basics, but I do have a few last things I'd like to mention.

Skyla sent:

Some commands will not print visual output to the Terminal in normal operation, such as the `cp` command, which is used to copy files and directories.

Skyla sent:

Many commands can use multiple arguments, handle multiple input types for arguments (e.g., file and/or directory paths), or have options that will have their own arguments.

Skyla sent:

Some options can, or must be, written in a long-form format (e.g., `ls --all [DIRECTORY]` is the same as `ls -a [DIRECTORY]`).

Skyla sent:

Options without arguments can often be provided together and in any order (e.g., `ls -la [DIRECTORY]` is the same as `ls -a -l [DIRECTORY]`).

Skyla sent:

And last, but certainly not least, you can almost always reference a command's format, options, and other arguments using the command `man [COMMAND]` (e.g., `man cp`) to view the provided command's manual page in the Terminal.

Skyla sent:

Hopefully that was not too much information! I know it seems like a lot, but it gets easier the more you use it. For today's tasks, I'll be sure to provide you with more details for any commands and programs you'll need.

Participant sent:
Sounds good! I'm ready to get started!

Skyla sent:

Excellent! I've gotten a request from our web development lead to set up a very basic web server and some other software for one of their new web developers, and I'd like you to help me **install** the necessary **software packages** on this Ubuntu Linux server.

Skyla sent:

Setting up new systems and installing software are some of the most common Systems Administration tasks.

Skyla sent:

When installing and managing software packages on a Linux system, you will most often do so with a package manager. Different distributions of Linux often use different package managers. The system we are working with today is Ubuntu Linux, which uses the **APT package manager**.

Skyla sent:

There are a number of different commands you can run in the **Terminal** that are used to interact with the APT package manager; however, for what we are going to do today, I'd suggest using the `apt` command.

Participant sent:
Got it!

Skyla sent:

Great! The first software package we need to install is **Caddy**. It is a fairly basic web server application.

Skyla sent:

Before we try to install it, let's make sure it's not already installed. To check if Caddy is installed, use the following `apt` command...

```
apt list -qi caddy
```

Skyla sent:

That `apt` command uses the `list` option, which is used to output a list of software packages, followed by two `list` options, `-q` and `-i`, and an argument, `caddy`, that modify what software packages are included in that list. The `-i` option limits the list to only installed software packages, the `-q` option removes additional notes and messages, and the argument limits the output to software packages that include the text `caddy`. The output will look like this if the **Caddy** software package is not installed...

Skyla sent:

```
playerone@workstation:~$ apt list -qi caddy
Listing...
playerone@workstation:~$
```

Skyla sent:

Okay, with that confirmed, let's install that web server!

Participant sent:
Sounds good!

Skyla sent:

Okay, as I mentioned earlier, we are going to use the APT package manager to install the Caddy web server. If APT had access to the internet, it could download software packages from online repositories managed by software and operating system vendors. However, this Ubuntu Linux server does not have internet access, so we will need to do a **local install** and provide the package manager with the software package.

Skyla sent:

I've put the Caddy software package in the **opt directory** (i.e., `/opt`). The Caddy software package file is named `caddy.deb`.

Skyla sent:

You can use the following `ls` command to display the contents of the opt directory and confirm the location of the Caddy software package file...

```
ls /opt
```

Skyla sent:

The command output should look like this...

Skyla sent:

```
playerone@workstation:~$ ls /opt
caddy.deb  curl.deb
playerone@workstation:~$
```

Participant sent:
Found it!

Skyla sent:

Perfect! Now I'd like you to install that Caddy software package for me.

Skyla sent:

There are a few different ways you could install the `caddy.deb` software package, but I'll suggest you use the `apt` command here. The `apt` command format for this will be...

```
apt [OPTIONS] [PACKAGE]
```

Skyla sent:

If you are not familiar with how to use the `apt` command to install the local software package, you could search something like 'install local deb package using apt' using a search engine on the web (e.g., Google, Bing, DuckDuckGo) or pull up the manual page for the `apt` command with the command `man apt` to help you look up what to do. I have to do this all the time!

Skyla sent:

If you do need to look it up on the web, be sure to do it from your computer and not the server. This server is not connected to the internet. 🚫

Skyla sent:

Additionally, since the command involves administrative changes to the system, you must run the command as a superuser. To do that, prepend the command you want to run with the `sudo` command (i.e., `sudo apt [OPTIONS] [PACKAGE]`).

Skyla sent:

When you use the `sudo` command, you will sometimes be prompted for your password. Just enter your password into the prompt and hit `Enter` or `Return` on the keyboard when you're done. (Note: Your password can be found on the Info Tab)

Skyla sent:

If you want help, just ask! I'm here to help you! 🌟

Participant sent:
Help, please!

Skyla sent:

Of course!

Skyla sent:

The command you need to run in the Terminal to install Caddy via `apt` using the `caddy.deb` software package is...

```
sudo apt install /opt/caddy.deb
```

Skyla sent:

You must start with the `sudo` command, as normal user accounts cannot make these changes to the system at their typical permission level.

Skyla sent:

The `install` option of the `apt` command indicates we want the package manager to install a software package. Then for the package argument, since we want to do a local install, we provide a path to the `caddy.deb` package (i.e., `/opt/caddy.deb`).

Skyla sent:

Once the install is complete, you can verify that the package was installed by running the command we ran earlier (i.e., `apt list -qi caddy`). It would give you the output, `caddy/now 2.6.4 amd64 [installed,local]` if the provided Caddy software is installed.

Skyla sent:

Once you've got Caddy installed, let me know so we can start your second task of the day!

Participant sent:
Caddy installed!

Skyla sent:

Excellent! Now we can install that other software package the lead web developer asked for.

Skyla sent:

You may have noticed another package in the `/opt` directory. The other software package file is named `curl.deb` and it installs the cURL tool. It is handy tool for interacting with web servers.

Skyla sent:

For your last task of the day, I want you to install the cURL software package. It will be very similar to installing the Caddy software package, so I'm sure you can handle it!

Skyla sent:

You will know you've installed cURL when you run the command `apt list -qi curl` and the output looks like this...

Skyla sent:

```
playerone@workstation:~$ apt list -qi curl
Listing...
curl/now 7.81.0-1ubuntu1.10 amd64 [installed,upgradable to: 7.81.0-1ubuntu1.13]
playerone@workstation:~$
```

Skyla sent:

Now, I've got to get to my meeting with the lead web developer to discuss what else we need to set up on this server. Thanks for all your help! 🌟

TRY CYBER

© 2023 Try Cyber - Sponsored by CISA