

Welcome to the

CIAB Remote Desktop System v3

Installation Guide

for Ubuntu 18.04 LTS

by brian mullan (bmullan.mail@gmail.com)

8/27/2019



Release Notes can be found at the end of this Document.

What is CIAB Remote Desktop System?

CIAB Remote Desktop (CIAB - Cloud-In-A-Box) was originally envisioned around 2008 after I had the opportunity from my then employer Cisco Systems to spend nearly 18 months on a Fellowship with a non-profit here in North Carolina that provides the networking connectivity (NCREN) to all of the schools in North Carolina.

At the time, cloud computing was just beginning and Amazon's AWS was practically

the only game in town. Having used AWS myself quite a bit by that time I tried to investigate how “cloud” could be used by K-12 schools as a possible low cost solution to the problems they faced such as:

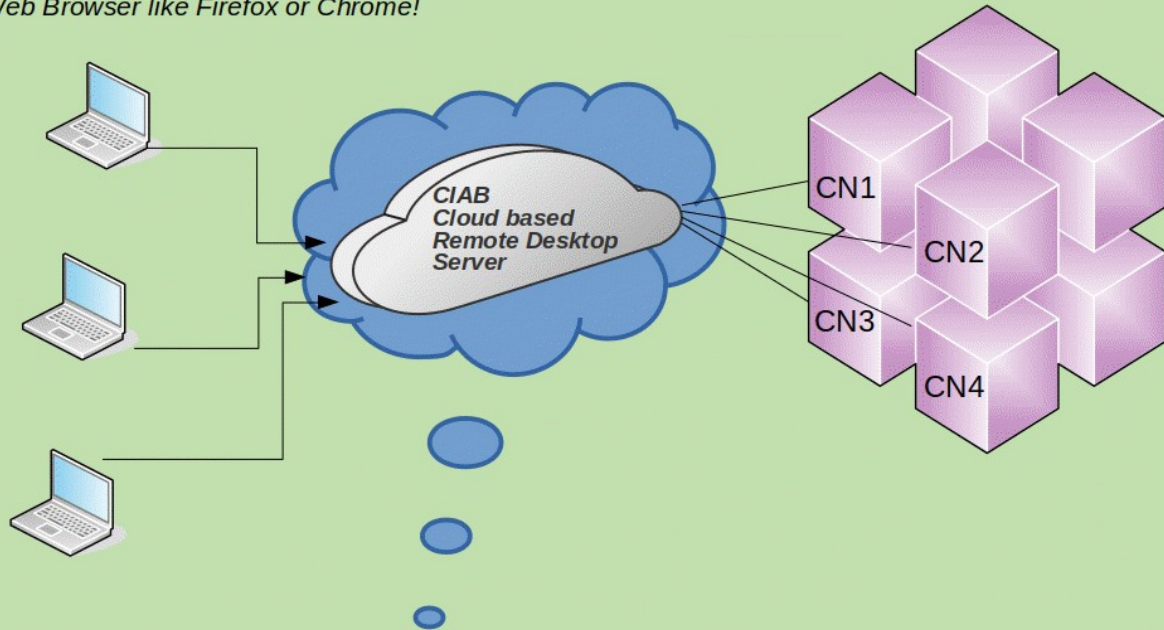
- lack of funds often prevent hiring top tech support or buying new equipment
- local inexperienced technical support which often-times consisted of a librarian, teacher or volunteers
- a hodge-podge of mixed old/new computers (desktop, laptops)

Today the available computers now also include mixes of chromebooks, tablets as well. Security & viruses on the student machines was a constant problem.

The above circumstances and combination of problems often created a frustrating experience for teachers, students and parents. So in 2008 I first starting thinking about how to bring together a Cloud based Remote Desktop solution that while not solving every problem, would try to adhere to the 80/20 rule of trying to solve 80% of the problems.

Cloud-in-a-Box (CIAB) Remote Desktop Architecture Overview

Access any Container Desktop using only a
Web Browser like Firefox or Chrome!



The CIAB Server is based on Ubuntu 18.04 LTS with the following installed & configured to provide an HTML5 Web based Management platform which integrates Guacamole, Tomcat8, Nginx, MySQL, FreeRDP, XRDP and LXD Containers to provide end-user access to LXD Container based Ubuntu 18.04 Desktop linux via Browser!

The amount of memory, disk drive space, operating system on the local computers **no longer matters** as the real User “desktops” are *remote* and the “server” they run on can be scaled in the “cloud” to as large as needed in size or number based on availability.

The school would only need decent Network connectivity in regards to speed & reliability.

For example, on AWS EC2 one of the larger Virtual Machine you can spin up approximates this:

| Instance Type | #vCPU | Memory (GB) | Storage (GB) | Network Speed |
|---------------|-------|-------------|----------------|---------------|
| m5d.12xlarge | 96 | 384GB | 4x900 NVMe SSD | 256bps |

Today there are lots of great IaaS (Infrastructure as a Service) Cloud providers including AWS, Digital Ocean, Hetzner and others.

If you were to install CIAB Remote Desktop on such an AWS server you would pay by the hour or month but as the above stats show you would be using a *very powerful* server to provide remote desktops to the users or students.

CIAB Remote Desktop Use-Case Benefits

1. Since any applications or databases used by the CIAB Remote Desktop users run on the remote server it doesn't really matter much how old or slow your local computing device is!
2. For an Admin, CIAB provides the ability to upgrade/delete/add or configure an application for many users by only doing so in one place not on dozens or hundreds of local computers.
3. Security. Regarding Security and/or viruses the remote desktop environments all are running on Linux. Security is managed in perhaps 1 or just a few servers versus again dozens or hundreds of local computers. Viruses... I'm not sure that there are any that affect Linux.

Also, **CIAB Remote Desktop uses HTTPS (SSL)** so the Browser connection to the remote desktop can be fully encrypted between the user and the Remote server providing the Desktop Environment.

4. For a business or a school, users can access their CIAB Remote Desktop while at work/school then go Home or travel and access the same Desktop using any HTML5 web Browser. Your remote desktop is always available to you from home or while traveling.
5. For Business, CIAB's CN1 container could be copied/cloned so that different containers could be targeted to different Business functional organizations such as Sales, Marketing, Operations, Engineering etc.

CIAB Remote Desktop could be useful for many different use-cases.

Besides the above benefits, if installed on one of your home computers you could access your Home Desktop from anywhere.

CIAB is useful even if you just wanted to use CIAB Remote Desktop on your own

laptop/desktop just to have multiple individual Desktops available to install/test or just work with.

Installing CIAB Remote Desktop

I've created and provided these scripts to completely automate installation of the CIAB Remote Desktop System for you onto an Ubuntu 18.04 Server whether that is a Physical local machine or VM, or a Cloud instance.

Also, there now several YouTube video's regarding CIAB I have created to help you with installation, configuration and use:

[CIAB Remote Desktop Part 1 - Installation](#)

and

[CIAB Remote Desktop Part 2 - Configuration and Use](#)

Before Starting the Installation Scripts

Some *assumptions*:

1. CIAB has been tested on Ubuntu 18.04 LTS. The only dependencies "may" be what version of Tomcat, mysql, nginx your Ubuntu has in its repositories.

In the scripts, "setup-guacamole.sh", "setup-nginx.sh" and "setup-ciab.sh" at the top of each script are defined Variables used to specify "versions" of software installed by each script.

2. You should install CIAB on a new Ubuntu 18.04 Server/Host, whether that is a physical server, VM or Cloud instance. The Server/Host should be running and you have access to it via SSH and sudo privileges on it.
3. You have NOT installed LXD on the server yet or if it was preinstalled then remove LXD. Executing both of the following will ensure that LXD is removed prior to the running of setup-ciab.sh as that will install SNAP LXD for you.

```
$ sudo apt purge lxd -y
```

```
$ sudo snap remove lxd
```

4. If using a cloud-server like AWS EC2 make sure you open ports 443 (https), 22 (ssh) & any other ports you may feel you want to open for other reasons.

This CIAB Remote Desktop installation process takes approximately 20-60 minutes (more or less depending on how "fast" your "server/Host" is).

By fast, we mean is it using SSD drives, does it have lots of memory and multi-core cpu (for cloud servers vCPU's)!

The CIAB Remote Desktop installation scripts provide lots of output on what the scripts are doing.

At times the scripts will prompt you as the installer to answer an install question.

To begin installation unarchive (un-tar or unzip depending on how you downloaded the files from the CIAB Github repository) the CIAB Remote Desktop installation scripts/files.

The installation scripts assume they all reside in the directory:

/opt/ciab

Change to that directory & make sure all the .SH files are executable.

```
$ cd /opt/ciab  
$ tar -xvf *  
$ chmod +x *.sh
```

Then start the installation

```
$ ./setup-ciab.sh
```

During installation, the SNAP version of LXD will first be installed in the Host/Server. LXD will be used to create the "system" containers that will run the CIAB User MATE Desktop system and the CIAB-GUAC Management system.

As part of this process the setup scripts will install the SNAP version of LXD in the Host/Server. When it does so you (the installer/Admin) will be prompted for

the configuration of LXD in the Host/Server.

You **must** answer the following questions with these responses.

NOTE: **for storage backend** for CIAB btrfs, dir or zfs usually works best so its your choice !

= = = = =

Would you like to use LXD clustering? (yes/no) [default=no]: **no**
Do you want to configure a new storage pool? (yes/no) [default=yes]: **yes**
Name of the new storage pool [default=default]: **default**
Name of the storage backend to use btrfs, ceph, cephfs, dir, lvm, zfs)
[default=zfs]: **<your choice see above NOTE>**
Would you like to connect to a MAAS server? (yes/no) [default=no]: **no**
Would you like to create a new local network bridge? (yes/no) [default=yes]: **yes**
What should the new bridge be called? [default=lxdbro]: **lxdbro**
What IPv4 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]: **auto**
What IPv6 address should be used? (CIDR subnet notation, "auto" or "none")
[default=auto]: **none**
Would you like LXD to be available over the network? (yes/no) [default=no]: **yes**
Address to bind LXD to (not including port) [default=all]: **all**
Port to bind LXD to [default=8443]: **8443**
Trust password for new clients: **<some secret password>**
Again: **<enter - some secret password again>**
Would you like stale cached images to be updated automatically? (yes/no)
[default=yes] **yes**
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]: **no**

= = = = =

Then four LXD containers will be created:

1. **ciab-admin**
2. **ciab-guac**
3. **cn1**
4. **ciab-mano**

First, container **CN1** is configured as a RDP enabled Remote Ubuntu MATE Desktop.

CN1 will be used for normal CIAB End-user Remote Desktop activities.

After the CN1 has completed installation of the Ubuntu-Mate desktop, CN1 will be used to clone/copy another LXD container named **CIAB-ADMIN**.

Next, a container named CIAB-MANO is created. A tool called LXDMosaic will be installed in the CIAB-MANO container.

LXDMosaic is a web based utility that provides Management and Orchestration (MANO) of LXD containers both locally and on remote LXD Host/Servers.

Reference: LXDMosaic - <https://github.com/turtle0x1/LxdMosaic>

Guacamole, NGINX, Tomcat, MySQL will be installed in the **CIAB-GUAC** Container.

You, the CIAB Admin, will only have to login via a Browser to Guacamole and configure Users and Connections.

To log into Guacamole you (the CIAB Admin) will point your web browser to the IP address of the Host/Server such as this:

`https://ip-address-of-host/guacamole`

The CIAB Installation process uses an LXD capability called **Proxy Device** to redirect HTTPS requests to the Host/Server to the CIAB-GUAC container and thus to Guacamole.

Guacamole uses the term "connection" to describe what desktop servers or LXD container desktop servers) to be able to reach.

Each UserID you configure in Guacamole as a Guacamole Admin will require a Guacamole LoginID & Password and also a LoginID and Password (re a linux userID and password) on each "connection" server (re LXD container such as CN1, CIAB-ADMIN) that the UserID is configured/allowed to access.

During the creation of the CIAB-ADMIN container, the SNAP version of LXD will again be installed but this time inside the CIAB-ADMIN container. This is done to allow LXD commands activated by you the Admin in the CIAB-ADMIN container to be processed by the Host/Server's LXD Daemon. This includes creation of other LXD containers

for CIAB Web Applications via the CIAB Web Applications installer (Icon will be on your CIAB-ADMIN Desktop).

However, this time the SNAP LXD will be configured using a “pre-seed” template configuration that the installation script provides so you do not have to re-enter and lxd config information.

After completing the previous configuration of LXD in the CIAB-GUAC container the rest of the installation will continue.

The LXD containers will appear & act like a separate servers even though they run on the same Server/Host. You could, as CIAB Admin, install different applications software in each LXD container for users to access via CIAB's Guacamole.

As all of the CIAB scripts execute you, the installer, will be prompted at times for input or to do a “next” action.

I hope most prompts will be self-explanatory.

Note: the only place the RDP protocol is utilized is from the CIAB Remote Desktop Web Proxy (re Guacamole) running in CIAB-GUAC LXD container and the Remote Desktop connection to the CIAB-ADMIN and CN1 containers.

Why RDP?

It is recognized that some use-case's may include not just Linux Desktop Servers but also Windows Servers. As RDP is the core protocol used in Windows Remote Desktop Connections (RDC) this allows greater flexibility in the overall CIAB Remote Desktop Architecture.

Again as a reminder, from the User to the Server/Host is HTTPS (TLS) encrypted communication capable because of our configuration of Nginx and a Self-Signed Certificate.

It is the responsibility of the CIAB installer to install/configure a LetsEncrypt Certificate if that is desired ! This is NOT an overly complex process and I've described it in one of the CIAB GitHub “issues” which you can follow.

CIAB Remote Desktop System Installation Steps

NOTE: the scripts have been written & configured to assume they are running from a directory named **/opt/ciab**. If you decide to do otherwise you will need to make modifications in most or all of the scripts to point to where you place all of the CIAB installation files

STEP 1

On the target Ubuntu 18.04 Host/Server create a new directory to hold all the installation files

```
$ sudo mkdir /opt/ciab
```

Make that directory "owned" by your UserID or the UserID of whatever acct you will login to on that "server"

```
$ sudo chown yourID:yourID /opt/ciab
```

STEP 2

Download the CIAB installation script files from Github:

<https://github.com/bmullan/ciab-remote-desktop>

Copy the provided archive to the target "server" and place it into /opt/ciab.

Example:

If you download the source using Github's ZIP file format the resulting archive will be called - "ciab-remote-desktop-master.zip"

So on your local PC/Laptop you would use SCP to copy that file to your target machine:

```
$ scp ./ciab-remote-desktop-master.zip yourID@ip-of-server:/opt/ciab/
```

If the unzip command isn't already installed on your system, then run:

```
$ sudo apt-get install unzip
```

After installing the unzip utility, if you want to extract to a particular destination folder, you can use:

```
unzip file.zip -d destination_folder
```

SSH - Log into that "server", and UnTar/Unzip the above file

```
$ ssh yourID@ip-of-server  
$ cd /opt/ciab  
$ unzip ./ciab-remote-desktop-master.zip (unzip if its a zip file)  
NOTE: if the unzip creates another sub-directory move all of the  
contents to /opt/ciab/ before proceeding!  
# make the bash scripts executable  
$ chmod +x ./*.sh
```

STEP 3

Start the installation:

```
$ cd /opt/ciab  
$ ./setup-ciab.sh
```

Note: you will be prompted several times during installation to either read a message then press enter or when installing Guacamole/MySQL/Tomcat/NGINX to input passwords for the MySQL root password and the Guacamole Database root password.

At the end of execution of setup-ciab.sh the script will prompt you what to do next!

Overall, installation can take from 20-60 minutes depending on how fast your "server" is (re does it have SSDs, multiple CPU cores, etc).

Note: We are using LXD/LXC **"un-privileged"** containers. You can later find the "rootfs" for those containers on the Server/Host located in the directory if you use our script as it installs LXD using the SNAP package manager. The SNAP installation of LXD puts the containers you create in:
/var/snap/lxd/common/lxd

Time to Reboot the Server/Host!

Rock and Roll - Time to try out your new Remote Desktops

IMPORTANT NOTE: *After the Host system is rebooted it can take 3-5 minutes for the CIAB Remote Desktop system to fully boot. Why? Because the “Host” itself has to boot, then the CIAB-GUAC container with its Ubuntu-Mate desktop, Guacamole, Tomcat, NGINX, MySQL has to fully boot and finally the CN1 CIAB Remote Desktop container with its own Ubuntu-Mate Desktop has to complete booting.*

- *So be patient and every 30 seconds or so hit refresh on your browser until you see the CIAB Remote Desktop login screen.*

Configuring CIAB's Guacamole

At this point everything is installed on the "server" (actually in the LXD containers on that Host/Server) but you still need to configure CIAB Remote Desktop by logging into the Guacamole Web Proxy “server”:

Using: **guacadmin** for the login ID and login Password

Point your HTML5 capable web browser to your "server"/Host using the following:

https://ip-of-your-HOST/guacamole

NOTE: even though Guacamole is installed in an LXD container named “CIAB-GUAC” running in your HOST/Server, the CIAB System installation software used LXD’s *Proxy Device command* to redirect any HTTPS traffic sent to the Host/Server to the LXD container CIAB-GUAC !

On first-time login to CIAB the installer/Admin needs to log-in as:

- **UserID: guacadmin**
- **Password: guacadmin**

which will present the Guacamole Configuration menu displayed in your browser.

In the upper right hand corner click on the UserID labeled **guacadmin** and then in the drop-down menu click "**settings**" then...

NOTE:

*At this point, follow the **CIAB Configuration and Use** video. It will show you step-by-step how/what you will configure for Guacamole Connections and Users in order to use CIAB as a Remote Desktop System!*

After Configuration of CIAB's Guacamole Connections & Users

You the CIAB Admin must create User Accounts on the target LXD container Ubuntu Mate Desktop systems.

The installation scripts will have already created a User Acct for yourself (as the Installer/Admin) both CIAB-GUAC and CN1.

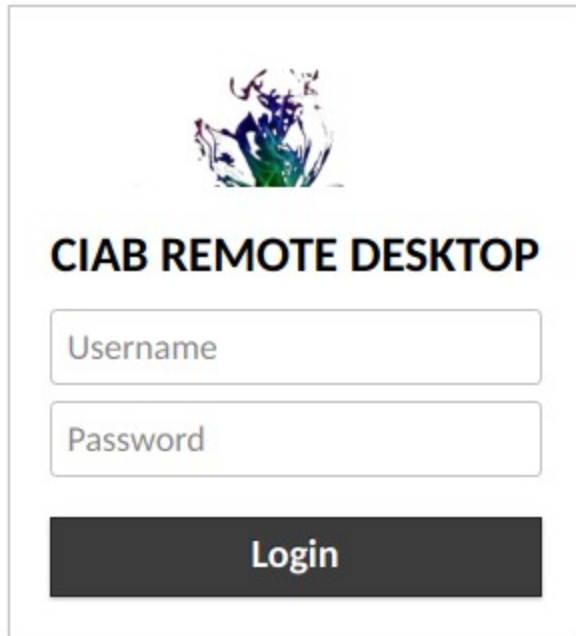
But if you need other User Accts in those containers you can do that later when you login to those containers after setup of the CIAB Remote Desktop. You will have already been give SUDO access IN those containers by the CIAB System installation scripts when you installed CIAB.

When you are your local PC/Laptop web browser to access CIAB you just need to point your Browser to:

https:<ip address of Host server>/guacamole

Because the installation scripts only install a "self-signed" Certificate for HTTPS, with Firefox or Chrome/Chromium you will see a warning message about an unsafe connection. In the future when you/Admin installs a valid Cert, like from **Lets-Encrypt** you will no longer see that warning.

So accept the connection and you will see the CIAB Remote Desktop Login menu:

The image shows a login interface for CIAB Remote Desktop. At the top center is a logo consisting of a stylized, colorful splash or burst of paint in shades of blue, green, and red. Below the logo, the text "CIAB REMOTE DESKTOP" is displayed in a bold, black, sans-serif font. Underneath the title, there are two input fields: the first is labeled "Username" and the second is labeled "Password", both in a light gray font. Below these fields is a dark gray rectangular button with the word "Login" written in white, bold, sans-serif font.

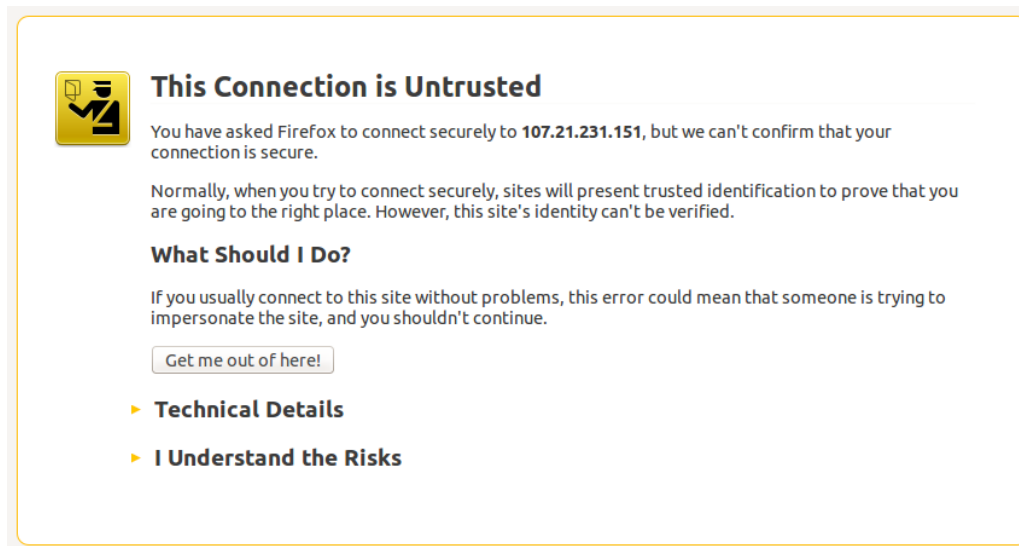
Picture: CIAB Remote Desktop Login menu

Enter a valid CIAB login ID and password and that User will be connected to their CIAB Remote Mate Desktop.

POST INSTALLATION - ERRATA

When accessing CIAB Remote Desktop with your Web Browser you will be presented with a screen something like the following (depends on what Browser you are using)... telling you that the Certificate presented is UNSIGNED.

NOTE: if you have bought a DNS Domain and registered that Domain with LetsEncrypt then CIAB can be configured rather easily to use the LetsEncrypt Certificate. In which case, you will not see any of the following!



Picture: Example Web Browser warning about Unsigned Security Certificate

That self-signed Security Certificate is created by the "setup-nginx.sh" script and although unsigned is safe for you to accept. If you are nervous examine the setup-setup-nginx.sh script & modify the NGINX section to suit yourself if you have obtained a valid "cert" & then reinstall nginx.

NOTE: even with the Self-Signed Certificate your HTTPS traffic from your local browser to the CIAB Remote Desktop will be fully encrypted.

Next Click on "***I Understand the Risks***" to continue. You will again be asked to confirm that you understand this is an unsigned certificate and again just accept it.

Timed One Time Password (TOTP) 2 Factor Authentication

The CIAB Admin only needs to do a couple simple things to turn on TOTP with CIAB.

NOTE: Do NOT do this unless all of your CIAB Users have a smart phone with the Google Authenticator application installed or they will NOT be able to complete login to CIAB !

First, log into the CIAB-GUAC container via SSH or by clicking on the Guacamole SSH “connection” you created when configuring Guacamole by following the CIAB Configuration and Use video.

In the terminal execute the following to download the TOTP file from the Apache Guacamole website:

```
$ wget http://apache.org/dyn/closer.cgi?action=download&filename=guacamole/1.0.0/binary/guacamole-auth-totp-1.0.0.tar.gz
```

Next un-tar the archive which will create a directory

```
$ tar -xvf 'closer.cgi?action=download&filename=guacamole%2F1.0.0%2Fbinary%2Fguacamole-auth-totp-1.0.0.tar.gz'
```

Next enter the guacamole-auth-totp-1.0.0 directory:

```
$ cd ./guacamole-auth-totp-1.0.0
```

Copy the guacamole-auth-totp-1.0.0.jar file to */etc/guacamole/extensions*:

```
$ sudo cp ./guacamole-auth-totp-1.0.0.jar etcguacamole/extenstons/
```

Finally, restart guacamole (or restart the CIAB-GUAC container).

```
$ sudo reboot
```


Now when you point your Browser to your Host/Server's IP address or your Domain Name you will find 2FA TOTP waiting for you to do the setup of 2FA on Google's Authenticator app.

The screen will look something like this:

Multi-factor authentication has been enabled on your account.

To complete the enrollment process, scan the barcode below with the two-factor authentication app on your phone or device.



After scanning the barcode, enter the 6-digit authentication code displayed to verify that enrollment was successful.

Authentication Code

Continue

Use your Phone's **Google Authenticator application** to scan in the "barcode" which will create an entry in your Google Authenticator labeled "Guacamole".

Read whatever is the current 6 digit code presented on your phone for Guacamole and enter it into the

screen form (***note that code changes every 15 seconds.*** Once you've done that you will next see where you need to sign the password etc

Printing

The CIAB Installation scripts **DO NOT** configure/enable Printing in the Host or the LXD containers as that is left up to the CIAB Installer to configure later.

However, there are several options I am aware of that you can choose from.

Both the Google CloudPrint Service or the IPP Everywhere technology cost nothing but some configuration time and in my opinion are best from a user experience point of view.

1. Guacamole's own print capability which downloads your printout as a PDF to your local machine which you then have to print the PDF to obtain the final printed output.
2. **Google's CloudPrint** service which most modern printers support! This allows remote Desktop systems to find & print ***directly*** to local CloudPrint enabled printers as you would on your local PC/Laptop.
3. Utilize the **IPP Everywhere** technology. This too can print directly to any **IPP Everywhere** printers with no intervening PDF involved.

Uploading and Downloading Files to/from CIAB and your Local PC

Upload Files

A CIAB user can upload a file **from** their local PC **to** their CIAB Desktop by dragging and dropping the file from their local PC ***to anywhere*** on their CIAB Desktop.

Those uploaded files will be placed under a folder named **ciab-drive** on your CIAB Mate Desktop.

Actually, the Uploaded files will be placed in a folder in your HOME directory labeled: ciab-drive/**GUACFS**/

Download files

If you want to “download” file(s) **from** your CIAB Remote Desktop **to** your local PC/Laptop you only need to drag & drop the file(s) into the folder:

ciab-drive/GUACFS/Download

which again you will find on your CIAB CN1 Mate Desktop. Once you do that *the downloaded file will be found in your local PC's Downloads directory/folder!*

Configuring an IPP Printer for CUPS on CIAB

The Internet Printing Protocol (IPP) is now a standard and most all modern Printers support IPP.

Step 1

Make sure you home/local printer has IPP enabled by using your printer's menu or web interfaces to enable IPP (usually enabled on **Port 631** and perhaps 80)

Step 2

On Home WiFi router configure PORT forward to forward "some port" to IP address of local Printer and its Port 631

Step 3

Then on Cloud server:

make CIAB Admin member of lpadmin group

Step 4 - create the CUPS IPP printer configuration

The following ***must be executed as sudo or root to use LPADMIN to add the local printer***

-p = printer_name = the name of the printer as it will appear on the Linux system in the Print dialog boxes

-v = ip_address = the PUBLIC IP address of home Wifi Router that the Printer is attached to via wifi
port_ID = the WIFI "source PORT" for the Port Forward config statement

-L = location of remote printer (any name you want to ID where the printer is located)

As CIAB Admin start your CN1 connection and when at the CN1 Desktop start a Terminal session.

In the Terminal at the Command line execute the following command to create an IPP Printer:

\$ sudo lpadmin -E -p printer_name -v http://ip_address:port_ID/ipp/ -E -m everywhere

Example:

```
$ sudo lpadmin -E -p my_printer -v http://47.61.218.28:6883/ipp/ -E -m everywhere
```

The above example assumes:

1. I want to call my home printer "my_printer"
2. that my Home Wifi router Internet address is 47.61.218.28
3. and that on that Wifi router I have already setup Port forwarding FROM 47.61.218.28 and Port 6883 to the IP address of my home printer's Internal Network IP address of 192.168.1.72 and my printer's default IPP Port of 631.

Now when an App on CIAB prints to the printer named "my_printer" it will be sent to my Home WiFi's Internet IP and port 6883 and get forwarded by my WiFi router to my actual printer IP and its IPP Port #631.

Now that you have configured the IPP printer in the CN1 CUPS you should be able to select & print to it so try to print to it. Remember, in whatever Application you use to print you will have to "select" the "printer_name" you configured as the Printer to print to !

Using Google's CloudPrint Service

Google provides a free service called CloudPrint.

Normally, to connect to a CloudPrint enabled Printer (most modern printers support Google CloudPrint) you have to have Google's Chrome (or the open-source Chromium-Browser) installed on the CIAB Remote Desktop container.

However, in the Ubuntu 18.04 repositories is a CloudPrint package you can install.

This is a Python implementation of the Google Cloud Print Connector **which does not require** the installation of the Chrome Browser on the CIAB Remote Desktop container. Refer to both the following. On the GitHub be sure to read through the "issues" for several tips on usage.

<https://pypi.org/project/cloudprint/>

and

<https://github.com/armooo/cloudprint>

Adding more Container Remote Desktop servers

There are two ways to accomplish this, manually using a Terminal and the LXD CLI or by using the newly introduced CIAB-MANO web application and doing it via the GUI.

Manual Process

If you would like to add more Container based Remote Desktop servers it is easy to do and a lot faster than when creating the first LXD container.

You may want to do this in order to install a different Desktop Environment (Ubuntu-Budgie or Xubuntu or Lubuntu) than Ubuntu-Mate. Or you may want to specialize the applications installed on one CIAB Desktop versus another!

To add more you need to “clone/copy” an existing LXD container to create more of them you can use the LXC “copy” command.

This will create an exact copy of the CN1 container & name it CN2:

First, stop the existing LXD container so you can clone/copy it (do this when no one is using it!):

```
$ lxc stop CN1
```

next clone/copy that container to a new container:

```
$ lxc copy c1 cn2
```

The above command would clone our CN1 container to a new container named CN2!

Remember, you can do this copy/cloning as many times as you need and will run well in your Cloud Server or DataCenter Host.

Restart the containers with the command:

\$ lxc start cn2 (or CN1)

Verify they are restarted & note their IP addresses so you can add the new container as a new Guacamole Remote Desktop “connection”. Use the following command:

\$ lxc list

You MUST also need to go back and add a new “connection” in the Guacamole Web Proxy configuration manager for the new Container CN2.

You can also use the LXD Copy command to “copy” an existing LXD container to a totally new Server/Host.

Copy/Cloning an LXD container locally takes perhaps a minute or two to complete.

You could repeat the above to create any number of new containers based off of an original “base” CN1 configured container.

Using the powerful capabilities of LXD/LXC you can also “migrate” (re move) an existing container such as CN1 to a totally different Server/host.

Refer to the [LXD Documentation \(ReadTheDocs\)](#) to learn more!

The installation scripts should set things up so that any future new users added to receive Audio (via PulseAudio).

My Own Example Demo Installation Info

To test the CIAB Remote Desktop Installation process out I tested it on both AWS EC2 and on Digital Ocean as well as on Hetzner Clouds.

Note: to understand the following assumes some knowledge of AWS EC2 is required

For AWS EC2 you can use [Canonical's AWS AMI finder](#) to pick an Ubuntu 18.04 appropriate size Server instance (re memory, SSD or spinning disk, #vCPU's etc) for your needs and your AWS "region".

I ssh'd into that instance, created a user acct for myself, gave myself sudo privileges, created the /opt/ciab directory & made my UserID the "owner" of /opt/ciab.

Then I logged out of the AWS server, used SCP to copy the "***ciab-remote-desktop-master.zip***" file to that server and into the /opt/ciab directory.

When that was complete I ssh'd back into the AWS server using my own UserID now and changed (\$ cd /opt/ciab) to the /opt/ciab directory.

Then following this document I began the installation process.

So... how long does Installation of everything take...??

From beginning to completion the entire process took 20-60 minutes (depending on which AWS Server instance – vCPU, Memory & whether it has SSD or spinning disks) to install everything on the above AWS Server/Host including the reboot!

But remember that included installing the full Ubuntu-Mate desktop environment in both the Host/Server and also in the initial CN1 LXD container (from which we later just made a copy/clone of CN1 to create further LXD containers).

After the AWS Server came back online I used Chromium (or Chrome) & HTTPS to access the CIAB Remote Desktop Web Proxy on the AWS Server, logged in as “guacadmin” and used the password “guacadmin” which are installed as defaults during Guacamole installation.

I then created connections, user accts for myself etc per this guide.

After that was complete I logged out of the Admin menu and logged back in as my own User ID and from there I could then access the Host itself and the CN1 container Desktop all through a Chromium browser.

CIAB Web Applications Installation and Use

After the CIAB Remote Desktop system has been installed and the Admin has configured the required connections in CIAB's Guacamole for the CIAB-GUAC and CN1 containers, the Admin can log into their account on the **CIAB-GUAC** container.

When they log in to the MATE desktop on CIAB-GUAC the Admin will find an Icon labeled “**CIAB Web Applications Installer**”. To install one or more CIAB Web Applications the Admin just needs to double click on that Icon and a menu will be presented where they can check the boxes for one or more applications to install.

After selecting those applications and clicking on “Install” the CIAB system will create a new LXD container for each selected Application. You will then be asked several questions which you **must** provide input to.

In each new LXD container the matching Application will be installed. The LXD container Name will be the same as the Application. For instance, if you install “Wordpress” the LXD Container will also be called “wordpress”

Note: that installation can take upto 5 minutes for each application so be patient and answer any questions you are prompted for.

If you screw up anything, you the Admin, can open a terminal or use CIAB-MANO tool and delete the associated LXD container for that Application and then just go through the process of re-installing it again.

To see a list or get the IP address of any of the installed CIAB Web applications & their containers you can use the CIAB-MANO tool.

Then you can reinstall it as the previous process described.

Steps to Implement a real/valid HTTPS/TLS Certificate for CIAB

CIAB's default installation implements what is called a "Self-signed Certificate". This is a cryptographic certificate and does enable encryption of your HTTPS connection to a remote CIAB system. However, because the certificate was generated locally and not by a Certificate Authority (CA) such as LetsEncrypt, users will always upon first accessing CIAB be presented with a "warning" message.

To eliminate that requires that you perform several actions:

Step 1 – Obtaining and registering a Domain Name

In order to obtain a real Certificate from LetsEncrypt (its free) one of the pre-req's is that you have a Registered Domain Name.

You can use a Domain Registrar such as GoDaddy or other to obtain a Registered Domain Name for you CIAB system.

There are a lot of different Domain Registrars and they all charge different annual fees for the Domain Name you pick. I used GoDaddy to register "ciab.ws" since I thought that intuitively implied "ciab work-station".

As part of that Domain Registration you will configure what's called a Domain A record entry. That will require you to enter the IP address of your CIAB system which will then be aliased on the internet by your new Domain Name. So for me entering "<https://ciab.ws/guacamole>" would take me to whatever server had my Domain Name IP address.

NOTE: if you install CIAB on a Cloud Server such as AWS or Digital Ocean. They all have a similar feature sometimes referred to as an "Elastic IP Address".

Normally without using an Elastic IP Address, everytime you reboot or restart your Cloud Server you would end up with a different IP address. Obviously, that would make your Domain Name access non-functional unless you logged back into your GoDaddy (or whoever) account again and change the IP address of the Domain Name to the new IP address of your CIAB Cloud Server.

To avoid having to do that with your Cloud Server... get an Elastic IP address and assign it to your server. *Elastic IP addresses usually are not free but they are cheap!* Now your Cloud Server can be rebooted, stopped/restarted and it will always come up with that same IP address.

After you Assign the Elastic IP to your Cloud Server you will need to log back into your Domain Registrar's website (GoDaddy or whoever) and edit your Domain Name to change its "A" Record IP address to be that Elastic IP address.

Now, once that is done and you have your Domain Name registered you can begin the process of obtaining a free LetsEncrypt Certificate for your CIAB installation.

Step 2 – Obtaining and installing a LetsEncrypt Certificate

CIAB used Tomcat, NGINX, MySQL for Guacamole.

To obtain and install a valid CA for CIAB is luckily fairly simple today because of a LetsEncrypt tool called CertBot.

Digital Ocean (the cloud company) has a nice document explaining the steps required and how to use CertBot to auto-magically get a Certificate from LetsEncrypt and to auto-configure CIAB's NGINX for you.

[Digital Ocean LetsEncrypt Guide for Ubuntu 18.04](https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04)

URL: <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>

Using the above guide, you as CIAB Admin first need to log into the CIAB-GUAC desktop and open a Terminal.

IMPORTANT NOTE: in the above Digital Ocean document the “**Step 2**” example tells you to edit NGINX's:

```
sudo nano /etc/nginx/sites-available/example.com
```

“example.com” is only an example file name.

Your CIAB NGINX file that needs to be edited is actually:

```
/etc/nginx/sites-available/guacamole
```

Before editing it, make a copy!

```
$ cd /etc/nginx/sites-available
```

```
$ sudo cp ./guacamole ./guacamole.bkup
```

Then edit the guacamole file (use nano or whatever text editor you want):

```
$ sudo nano ./guacamole
```

For CIAB you will search for:

```
server {
```

```
listen 443 ssl;
```

```
server_name localhost;
```

and change “localhost” to your new Registered Domain name!

In my own case I changed “localhost” to:

```
server_name ciab.ws;
```

Remember to save your change!

NOTE: Digital Ocean's document “**Step 3**” must be performed in the Host/Server and

in the CIAB-GUAC Container !

If you do NOT open these ports then CertBot will Fail to obtain & install a LetsEncrypt Certificate into your CIAB NGINX.

With CIAB you will also see the HTTPS port (443) open as well as Port 22 (SSH) on the Host/Server

In the Digital Ocean Document's "**Step 4**" where it's example says:

```
sudo certbot --nginx -d example.com -d www.example.com
```

While still in a terminal *in the CIAB-GUAC container* you will execute the above but change "example.com" to your own Registered Domain Name!

The Digital Ocean Guide describes the Questions LetsEncrypt will ask you to answer.

IMPORTANT NOTE: If for any reason CertBot "fails"... be aware that LetsEncrypt ONLY lets you retry 4 times in any one week to obtain and install a valid Certificate.

If CertBot reports a failure... then RECHECK /Verify that you have:

- opened the port 80 and 443 (the Digital Ocean Guide tells you want Ports)
- that you entered your Registered Domain Name correctly on the CertBot command
- that you edited and save the NGINX sites-available/guacamole file correctly

Before you try again. If you fail 4 times you will have to wait a week to try again.

If CertBot fails twice i'd recommend reading the following which describes how to use LetsEncrypt's test capability to test your CertBot out without incurring any more Failures.

Once that works correctly then retry the actual command to get the Certificate & install it.

<https://letsencrypt.org/docs/staging-environment/>

i used it on Ubuntu 18.04 with guacamole, mysql, tomcat, nginx and had nginx configured with a Certificate in 5 minutes.

Anyway, thought perhaps if you hadn't used it before this would interest you since you can just use the default nginx sites-available/default file and certbot reconfigures it for you.

NOTE:

CertBot automatically takes care of auto-renewing the cert without you having to create any cron jobs etc !!

That's all there is to it !

From there you might want to log into any/all the Connections and create more User Accounts for other Users.

NOTE:

➔ For each new User you create in Host or an LXD container you will need to subsequently create a new UserID in CIAB's Guacamole for that User as well. For each CIAB Guacamole User account created remember to check the box in the Guacamole configuration screen for each "Connection" you want each Guacamole User to have access to (can be one or more of the available Connections)!

FINAL NOTES

Remember this is an attempt to demonstrate Guacamole HTML5 Remote Desktop capabilities to an Ubuntu Linux system using only a Browser from the Client PC, Laptop, Tablet (or phone).

With the availability of CIAB Remote Desktop System everything is installed & run from LXD containers.

This will be very useful because you can create your own LXD Image Repository and publish your CIAB containers to that Repository.

You can also use the CIAB-MANO tool to copy those Containers to a different LXD Host/Server on the same or a different remote Cloud environment and instantly have a new running environment there to use.

Then for future new installations all you'd have to do is install a server somewhere, install LXD on it and then use the CIAB-MANO to copy/clone your existing CIAB containers to it.

Please feel free to contribute fixes/enhancements to the Github files if you are able!

This is just a beginning. But my hopes are that the use of LXD containers could eventually enable Guacamole to support a large number of LXD container desktop targets running off a single (or cluster) of powerful Cloud or In-House servers.

Thanks for checking this out...

Brian Mullan (bmullan.mail@gmail.com)

Release Notes

CIAB version 3 introduces the following improvements.

CIAB-MANO (MANO is an acronym for Management and Orchestration) utilizing LXDMosaic's Browser based LXD container management system.

CIAB-MANO is installed and run in its own CIAB LXD container aptly named "ciab-mano". The installation also adds a new Launcher Icon onto the CIAB Admin's ciab-guac Ubuntu Mate desktop. Clicking on that Icon will present the CIAB Admin with the LXDMosaic main menu.

CIAB-MANO adds a completely GUI driven way to manage/orchestrate your LXD containers including:

- Create (on local or remote LXD Hosts/Servers)
- Delete (on local or remote LXD Hosts/Servers)
- Start (on local or remote LXD Hosts/Servers)
- Stop (on local or remote LXD Hosts/Servers)
- Freeze (on local or remote LXD Hosts/Servers)
- Unfreeze (on local or remote LXD Hosts/Servers)
- Migrate (to local or remote LXD Hosts/Servers)
- Copy (to local or remote LXD Hosts/Servers)

Documentation for LXDMosaic can be found here:

<https://github.com/turtle0x1/LxdMosaic>

NOTE: *when installing CIAB you do NOT have to execute the "Install Script" section of the above Documentation's URL ! CIAB will do that for you automatically during installation.*

Prep Work to enable your CIAB-MANO to manage Local/Remote LXD Hosts/Servers over the Network/Internet

For ***each*** Local/Remote LXD Host/Server you want to have managed by CIAB-MANO you need to enable access from the network on your LXD hosts first.

You can do this by logging onto each of your LXD Hosts/Servers and execute the following (make sure to change the password from "some-secret_string").

```
$ lxc config set core.https_address [::]  
$ lxc config set core.trust_password some-secret-string
```

NOTE: *whatever you use for "some-secret-string" will be used on each local/remote LXD Host/Server*

On your CIAB system where the CIAB-MANO (re LXDMosaic) is installed you will also need to perform the following:

1. ssh into the CIAB Host/Server **OR** as CIAB Admin log into ciab-guac and open a terminal
2. Enable access to/from the network for LXD on your CIAB Host by executing the following.
Use the same “some-secret-string” as above!

```
$ lxc config set core.https_address [::]
```

```
$ lxc config set core.trust_password some-secret-string
```

After you have done this you, the CIAB Admin, can log into CIAB again using your Browser and click on your “ciab-guac” connection and access your Ubuntu Mate Desktop there.

On that Desktop you will find a new Launcher ICON labeled “**CIAB Container Orchestrator**”.

If you click on that Icon, a browser will appear and you will be taken to the CIAB-MANO Web Application (LXDMosaic) running in its own LXD container called “ciab-mano” on the CIAB Host/Server.

V 2.3 also upgrades XRDP from v0.9.5 to 0.9.9.1 to fix a Drive Redirection Bug that affects CIAB User’s ability to Upload/Download files to/from their CIAB Remote Desktop.

CIAB version 2.2 introduces the following improvements.

- *TOTP* (Timed One Time Password) 2 Factor Authentication (2FA) is now an optional capability which the CIAB admin can easily install and it will automatically activate for use by CIAB users logging in afterwards.

NOTE: users will have to have a TOTP compatible application on their smart phones in order to use TOTP. Google Authenticator works well for this and is available on Android and iPhone.

- This Installation Guide now has an added section describing the 4 steps required to create an IPP (Internet Printing Protocol) Printer in the CN1 CUPS (Common Unix Printing System) so users can print directly to their local printers (if those printers support IPP).

Any CIAB Web-Apps installed by the CIAB Admin from the Menu available on their CIAB-GUAC container Mate Desktop now get installed in LXD Containers as “peer containers” on the 10.x.x.x private network that CIAB-GUAC and CN1 containers are on.

NOTE: This is a change from CIAB’s previous use of “nested” containers for the CIAB Web-Apps and was driven by a bug in upstream Apparmor concerning “nested” Apparmor profiles. That bug may be a while before it is fixed so a decision was made to make this change.

Now as Admin, from the Host/Server you can find/see all installed applications and the CIAB-GUAC and CN1 containers and their IP addresses by executing:

\$ lxc list

Lastly, this document now also contains a section titled “***Steps to Implement a real/valid HTTPS/TLS Certificate for CIAB***”.

This section provides information and a Guide for how to obtain & install a valid Certificate from a Certificate Authority (re LetsEncrypt) for use for HTTPS/TLS access to your CIAB installation.

CIAB version 2.1 introduces the following improvements.

File Uploads/Downloads to/from the CIAB User's Remote Desktop

Both file uploads and downloads now work between a Users local PC and their CIAB Remote Desktop system.

To UPLOAD a file To the User's CIAB Remote Desktop simply **Drag** the file from their local PC/laptop and **Drop** it anywhere on their CIAB Desktop.

They will find the uploaded file by clicking on **ciab_drive** then **GUACFS**

To DOWNLOAD a file To the User's CIAB Remote Desktop simply **COPY** the file they want to download to their local PC and PASTE it into the CIAB **Desktop** folder:

ciab_drive/GUACFS/Download

They should see a message in the lower left corner of the screen about the file being transferred. When its complete the User will find the downloaded file in their Local PC's Download directory.

CIAB now requires one login for User to access CIAB Remote Desktop

Entry of a Login ID and Password is now only required one time for both CIAB's Guacamole and the CIAB Remote Desktop connections.

If a CIAB User only has a single Remote Desktop connection configured for them they will be connected to that Desktop immediately after entering their User ID and Password.

1. If a CIAB user has only one CIAB Guacamole “connection” configured for them by the CIAB Admin then after entering their Login ID and Password they will automatically be presented with that connection's Ubuntu Mate Desktop. If however, they have more than one

connection configured then they will be presented with a menu from which they can choose which connection (ie Desktop) they wish to connect to.

2. A CIAB Remote Desktop user is now only required to enter their CIAB Remote Desktop system UserID and Password one time to gain access to any CIAB Remote Desktop that the Admin has configured them to have access to. Previously the users were required to enter their UserID and Password once to log into CIAB's Guacamole, then after clicking on one of their Admin permitted Remote Desktops they were required to enter their UserID and Password again to access the Ubuntu linux running in the LXD container and its Ubuntu Mate desktop. This second login is no longer required!

IMPORTANT NOTE: This assumes the CIAB Admin has done 2 things during installation of the CIAB Remote Desktop system when configuring each “connection” within CIAB's Guacamole.

PARAMETERS

Network

Hostname: 10.43.153.173

Port: 3389

Authentication

Username: \${GUAC_USERNAME}

Password:

Domain:

Security mode: RDP encryption

Disable authentication: ☐

Ignore server certificate: ☐

Remote Desktop Gateway

Hostname:

• First, when configuring the CIAB's Guacamole “connection” for a Desktop the Admin must change the following the Username and Password entries as shown in the Authentication section of the picture above!

- Set the Authentication field Username to: **`${GUAC_USERNAME}`**
- Set the Authentication field Password to: **`${GUAC_PASSWORD}`**

NOTE: Guacamole prevents you from seeing the text you enter in the Password field

unless you 1st click on the little Lock icon to the right of this field.

- Second, to use this feature/capability, the CIAB Admin **must** configure the **same** UserID and Password in CIAB's Guacamole as in the LXD Ubuntu containers those Desktops run in.

IMPORTANT NOTE:

When you initially create a NEW User account in Guacamole you should:

Assign them a **UserID** and **Password** and click on the *Permission Box* to allow “that” UserID to be able to change their own Guacamole Password.

Then ***create that same UserID and Password on the CN1*** container using normal Linux/Ubuntu method/command in a Terminal:

\$ sudo adduser UserID

After you have done that the UserID user can use their browser to log into Guacamole, select CN1 and log into their CN1 Ubuntu Mate Desktop.

When they get to their Ubuntu Mate Desktop they should perform the following Process to Change their Guacamole and CN1 Password:

1. The User should go into Guacamole by pressing **CTRL, ALT** and **SHIFT** keys on the **LEFT side** of their keyboard ***SIMULTANEOUSLY*** which will open a Slide-Out window.
2. *Click on their UserID/Name* at the top of that Slide-Out window
3. Select “**Settings**”
4. Enter their Current Password (that you assigned them)
5. Then they enter what they want their new Password to be
6. Press **CTRL, ALT** and **SHIFT** keys on the **LEFT side** of their keyboard ***SIMULTANEOUSLY*** again to **CLOSE** the Slide-Out window and return to their CN1 Ubuntu MATE Desktop.
7. They then need to open a Terminal
8. At the Terminal Prompt they need to change their Ubuntu Password to match

what they changed their Guacamole Password to by using the command:

\$ passwd

After doing this, they should log out of the CN1 Mate Desktop and when entering Guacamole again but using the new Password they set for themselves.

NOTE: *capability to use a one-time login to CIAB Guacamole to connect to one of the CIAB Desktops is **Optional!***

In some use-cases you may want/need a separate LoginID and Password for the CIAB Guacamole and one or more target remote desktop servers.

Example: if there was an existing Windows Server and you do not control the Login ID and Passwords of users on that machine.

If that is the case, then you will not want to configure the CIAB Guacamole “*connection*” to that machine with any entries in the Authentication Section for “Username” and “Password”.

NOTE: In a future version of CIAB Remote Desktop System I hope to figure out how to integrate some sort of single sign-on (SSO) applicaiton capability for both CIAB Guacamole and for the CIAB Remote Desktop users.

CIAB version 2.0 has many changes and improvements.

1. Guacamole v1.0.0 has now been integrated/implemented. Guacamole v1.0.0 introduces major new features/capabilities to Guacamole such as:
 - Support for User Groups
 - Multi-factor authentication with Google Authenticator / TOTP
 - Support for RADIUS authentication
 - Support for creating ad-hoc connections
 - Support for renaming RDP drive and printer
 - Cut & Paste for text-only (no pictures) now works as it normally would on a desktop
 - Configurable terminal color schemes
 - Optional recording of input events
 - SSH host key verification
 - Automatic detection of network issues
 - Support for systemd

- Incorrect status reported for sessions closed by RDP server
 - Automatic connection behavior which means Guacamole will automatically connect upon login for users that have access to only a single connection, skipping the home screen.
2. All supporting applications, including Guacamole, Tomcat, NGINX, MySQL etc are now installed in an LXD container (CIAB-GUAC).
 3. A new capability utilizing the recently added LXD Device Map feature, is now automatically configured when ciab-desktop has installation is complete. It will map Port 443 (re HTTPS) on the Host Server to Port 443 in the CIAB-GUAC LXD container. After this, any remote Browser pointed to your Cloud or VM Host Server IP address & Port 443 will be redirected to the LXD CIAB-GUAC container's Port 443 where it accesses Guacamole so initial Admin setup with Guacamole can be accomplished.
 4. Since the CIAB's Guacamole container and any Desktop containers (re CN1) are all on the same internal private 10.x.x.x network subnet CIAB's Guacamole will be able to let users access any other LXD container cloned from CN1 that you create (assuming you configure CIAB's Guacamole with "connections" to all containers).
 5. An extensive collections of Web Applications have been included for selection by the CIAB Admin. These applications are especially selected as best-of-class in open source for categories such as and the CIAB Admin can install them via a convenient GUI application.

These applications will be installed as individual, LXD containers. Each of the CIAB Web Application containers will be attached to the same 10.x.x.x private network that the CIAB-GUAC management container and the CN1 user MATE Desktop Container are attached to. This will enable any validated CIAB Mate Desktop user :

- *Enterprise Resource Planning (ERP)*
- *Project Management*
- *Content Management Systems (CMS)*
- *Social Media systems*
- *eCommerce Systems*
- *Learning Management Systems (LMS)*

- *IT Management systems*
- *Blogging systems*

The implementation LXD containers for these Web Applications greatly reduces their Security exposure footprint!

This is due to the fact that the Web Applications by default are ONLY accessible by validated CIAB Desktop users and ONLY on the private 10.x.x.x network. Those applications, by default, are NOT accessible from the Internet although the applications themselves have access “to” the Internet although at the discretion of the CIAB Admin they can change that to where Internet Users could be allowed access to one or more of the installed CIAB Web Apps.

6. ***Sound/Audio now works !!! In any of the LXD CIAB Desktop containers.***