Welcome to CIAB Remote Desktop Installation Guide

for Ubuntu 16.04 LTS

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

5/10/2016

What is CIAB Remote Desktop

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.

CIAB Remote Desktop only requires a working HTML5 web browser!
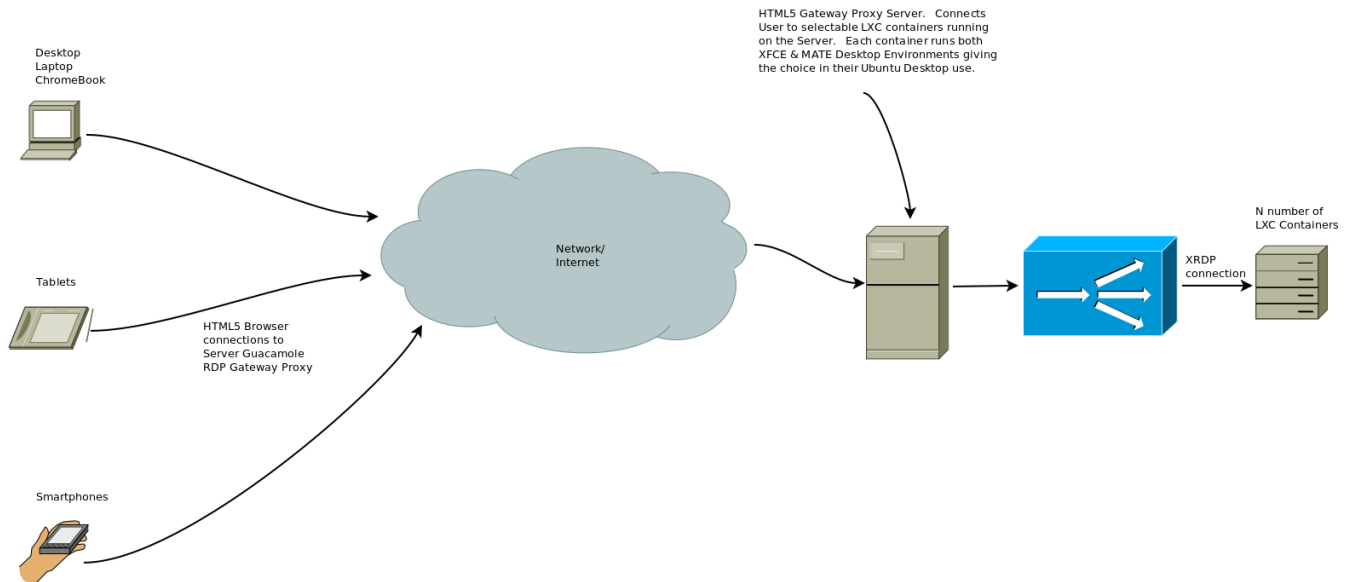
The amount of memory, disk drive space, operating system on the local computers **no longer matters** as the real User "desktops" are *remote* and the "server" they run on can be scaled in the "cloud" to as large as needed in size or number based on availability.

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

For example, on AWS EC2 the largest Virtual Machine you can spin up today is an "**instance**" called "**d2.8xlarge**":

Instance Type	vCPU	Memory (GiB)	Storage (GB)	Network Speed	Physical Processor
d2.8xlarge	36	244GBytes	24 x 2TByte	10 Gigabit	Intel Xeon E5-2676 v3

Today there are lots of great IaaS (Infrastructure as a Service) Cloud providers including AWS, Digital Ocean 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 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… to upgrade/delete/add or configure an application only requires 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 is fully encrypted between the user and the Remote server providing the Desktop Environment.

4. For a school, students can access their CIAB Remote Desktop while at School or Home just using a web Browser. Do homework at home or at school just using a browser! For non-students, your remote desktop is always available to you from home or while traveling.

Beyond schools, CIAB Remote Desktop could be useful for many people.

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

But 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 3 scripts to completely automate installation of the CIAB Remote Desktop for you onto an Ubuntu 16.04 server (local/cloud or a VM).

Before Starting the Installation Scripts

Some assumptions:

1. CIAB has been tested on Ubuntu 16.04 LTS. The only dependencies "may" be what version of Tomcat, mysql, nginx your Ubuntu has in its repositories.

    In each appropriate script, "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. A new "server" or VM is already installed, its running and you have access to it and sudo privileges on it.

3. 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.

    NOTE: It is recommended for end users to utilize the Chromium (or Chrome) web browser and not Firefox.   The reason for this is that Firefox has exhibited worse performance in regards to remote viewing of video/audio over the HTML5 connection than Chromium/Chrome.

This CIAB Remote Desktop installation process takes approximately 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!

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.

Examples:

When the script installs NGINX, if you are installing on a Cloud server you may get this prompt:

> *Command may disrupt existing ssh connections. Proceed with operation (y|n)?*
>
> *Just respond with "y" for yes… in my testing it has no effect on your session.*

After execution of the first three scripts (setup-guacamole.sh, setup-nginx.sh and setup-ciab.sh) the Guacamole Remote Desktop will be installed on the Server and you will only have to login via a Browser to Guacamole and configure Users and Connection (what desktop servers to be able to reach) information.

Also included are two optional extra scripts.   These scripts use LXD/LXC container technology to add 2 more Remote Desktop servers to that same Server/Host (see optional step #7).   If you choose to try the use of LXD container remote desktop out also then those scripts will again install Ubuntu-Mate desktop environment in each container.   Each container will appear & act like a separate server even though they run on the same Server/Host.   You can as admin install different software in each LXD container for users to access via 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.

There are 7 basic steps involved to install CIAB Remote Desktop onto your Server/Host and create 2 LXD/LXC containers.

This process will install on the Server/Host:

- CIAB Remote Desktop HTML5 web proxy is based on the great Guacamole project (see www.guac-dev.org) to enable connections using an HTML5 compatible browser

- the Ubuntu-MATE desktop environment

- mysql

- nginx

- tomcat8

- xrdp v0.9.0

  note that the xrdp found in the current Ubuntu 16.04 repositories is a much older version v.0.6.0

The **_optional Step #7_** will also install:

- LXD/LXC (to support LXD container creation/management)

- If you decide to experiment with LXD/LXC containters the Installation process
  will also create two LXD/LXC containers named CN1 and CN2 on the Server/Host

- xrdp/x11rdp in **both** CN1 and CN2

- in CN1/CN2 - the Ubuntu-MATE desktop environment

- in CN1 & CN2 .. a User acct w/SUDO privileges for you (the Installing User)
  so you can later log in and do admin activities like add more users.

Note: the only place the RDP protocol is utilized is from the CIAB Remote Desktop
Web Proxy running on the Server/Host to allow a Desktop connection to the Host
itself and from the CIAB Remote Desktop Web Proxy to the two LXD containers which
also are running on the Server/Host.

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 only 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 (TSL) encrypted
communication!*

CIAB Remote Desktop Installation Steps

NOTE: the scripts have been written & configured to assume they are running from the directory
/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 16.04 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

Copy the provided archive *"install-all.tar.gz"* to the target "server" and place it
into /opt/ciab.

     $ scp ./install-all.tar.gz yourID@ip-of-server:/opt/ciab

SSH - Log into that "server", and UNTar the above file

```
$ ssh yourID@ip-of-server
$ cd /opt/ciab
$ tar -xvf *
```

STEP 3

Install the Guacamole HTML5 Proxy Server onto the "server"/Host

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

Note: you will be prompted to input 3 passwords during this step.
   • A password for mysql
   • A password for nginx
   • A password for the Guacamole Web Proxy Server GUACADMIN acct

STEP 4

Install the NGINX Reverse Proxy Server onto the "server"/Host.   This will enable
HTTPS/TLS encryption for the user Web Browser connections to the Host & CIAB Remote
Desktop

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

Note: After step #4 you should reboot for the first time then log
back in, and again change directory to /opt/ciab and continue with
Step #5.

STEP 5

Note: this step "may" take 30-45 minutes depending on how fast your "server" is (re does it
      have SSD, multiple CPU core etc)

Install the CIAB Remote Desktop onto the "server"/Host

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

OPTIONAL STEPS 6 & 7 - only required IF you want to experiment with LXD container based Linux Remote Desktops also

STEP 6 (optional)

Setup/Install the LXD/LXC container system

NOTE: do NOT use sudo to execute the following it will prompt when required !

> *$ cd /opt/ciab*
> *$ ./setup-lxd.sh*

STEP 7 (optional)

This step will create the 2 LXD containers on your "server" that we will use for our demonstration of CIAB Remote Desktop in LXD/LXC containers.

These containers will be "privileged" containers. The 2 containers will be named "cn1" and "cn2".

> Note: We are using LXD/LXC "Privileged" containers. You can later find the "rootfs" for those containers on the Server/Host located in the directory:

> ### */var/lib/lxd/containers/*

> If you decide to investigate further you will have to have SUDO privileges to access that directory and any subdirectories!

Using your regular (non-root/sudo) login ID:

> *$ cd /opt/ciab*
> *$ ./setup-containers.sh*

> **Note:** this may take 45 minutes again depending on how fast your "server" is (see above note).   It takes this long because the script is again installing the Ubuntu-Mate Desktop Environment into the Containers.

When this script finishes you will see the output of the "lxc list" command which will show you the 2 LXD containers status and their IP addresses.

Write down those 2 IP addresses as you will need them later when configuring CIAB Remote Desktop as an "admin" to create "***Connections***" for users to access those two containers from their local Web Browser.

If you forget to write the IP addresses down you can always find them again by executing the command **"lxc list"** from a terminal prompt

BOTH containers CN1/CN2 will have the *Ubuntu-MATE desktop environment* installed into them with YOUR user account created & having sudo privileges IN those containers (again so you can be admin in them in the future).

Time to Reboot the Server/Host one last time!

Rock and Roll - Time to Try your new Remote Desktop(s) out

> Note: The reboot it can take up to 3-5 minutes because of all the web servers
> etc that need to load & initialize (especially on a Cloud service like AWS or
> Digital Ocean). So be patient & every couple minutes just retry the HTTPS
> address again until you see the Guacamole Web Proxy login box.

Configuring Guacamole

At this point everything is installed on the "server" but you still need to
configure CIAB Remote Desktop by logging into the Guacamole Web Proxy "server":

Using: **guacadmin** for the login ID
        use the password you input during Step #3 for Guacamole's *installation*

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

> **https://ip-of-your-server/guacamole**

You need to 1st login as guacadmin/guacadmin which will present CIAB Remote Desktop
management menu displayed in your browser.

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

Step 1:
Change language preference for the admin account.

Click on PREFERENCES

- change the Display Language to what suits you

Step 2:
**Define what Desktop Server "connections" you have setup for users to connect to.
In our demo installation the minimum will be 1 connection for the Host/Server
itself and "optionally" a "connection" for each LXD container CN1 and CN2 if you
chose to do that during installation.**

Click on CONNECTIONS

- add a new connection

Picture #1: Example of HOST Connection configuration

Picture #2: Example of CN1 Connection configuration
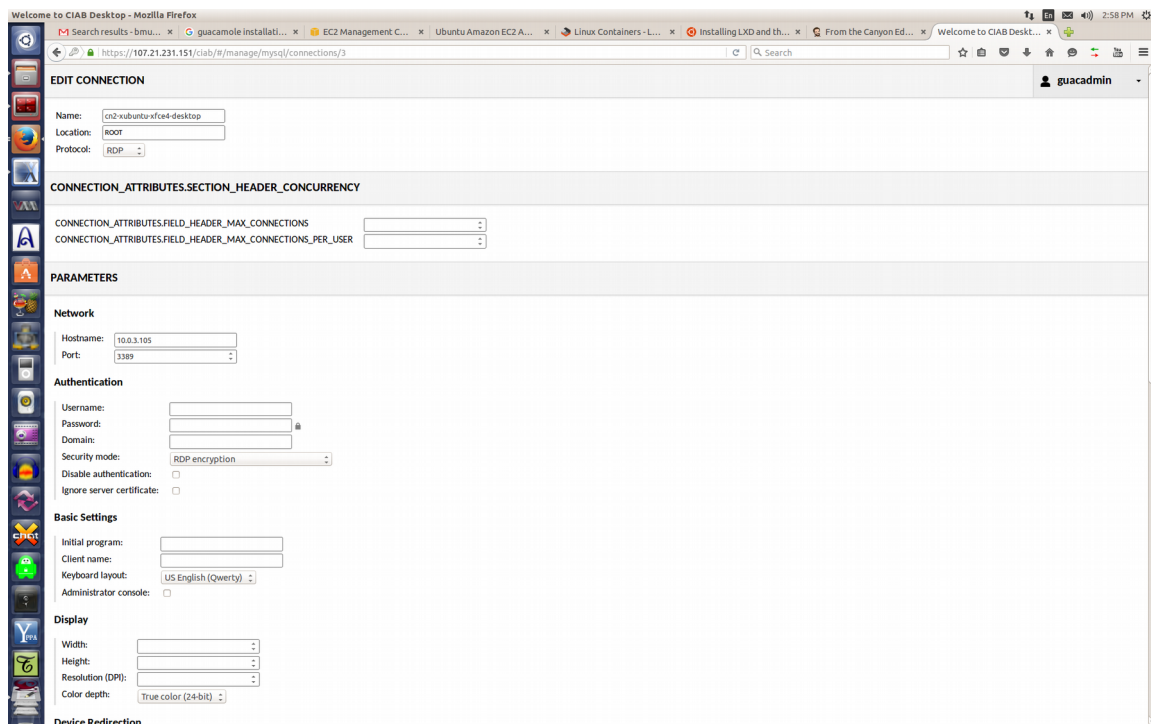
Picture #3:   Example of CN2 Connection configuration

Repeat Step 2 for EACH of the two LXD containers if during initial installation you decided to try that out.

For EACH connection you create:

Enter a meaningful "name" for the connection!

For example, you might just want to call them "Host-Server", "CN1-Ubuntu-Mate-Desktop" and CN2-Ubuntu-Mate-Desktop" for simplicity & easy identification. However, you might decide to have each Desktop to have different sets of applications installed later from a user functionality perspective… like a "science", "general" and "history" (whatever your use cases are) Desktop setup??

Change the *type connection* from VNC to **RDP**

For the "host/server" connection enter 127.0.0.1 and 3389 for the Port

For the LXD container (CN1/CN2) connections – enter the IP addresses you wrote down that were displayed during installation of the CN1 & CN2 containers. They will probably be something like **10.x.x.x**

for *Encryption*... **select RDP Encryption**

for *Keyboard* select what you use (qwerty english is default)

for *Screen Depth* **select 24 bits** (I had a problems w/32 bit)

For now that's all you need so at **scroll to the bottom & select SAVE**..!

Step 3

Click on USERS

Add a new User ID for yourself and any others including possibly a "guest" user.

> ***IMPORTANT NOTE*****:**
>
> **The Guacamole Web Proxy User IDs you enter here are SEPARATE & DISTINCT from the Linux User Acct IDs in the "server" and the 2 LXD containers !!**
>
> **These IDs are only used to allow access to the CIAB Remote Desktop web proxy system using their HTML5 compatible Browser**.

Remember for EACH user to check the boxes at the bottom for EACH connection you want to allow them to Connect TO !!

As admin, *you may give them access to one or many Connections* as you may later have dozens of servers they could connect to.

> **TIP**: you may want to also check the box to let them change their own password!

After successful login/password the users will get a "***Connections***" menu (configured by the admin) where they can click on any Connection, you as the Admin, have setup for them.

> ***Note:*** <u>you can</u> make the Login ID and Password in the Guacamole Web Proxy <u>different or the same</u> as the Linux User's ID and pwd you created on the Host or in the containers (cn1 and cn2).

First, create a User for yourself!

If you are going to do Guacamole Web Proxy admin duties later you might want to **check all the boxes** under PERMISSIONS!

Picture #4:   Adding a new User configuration.

NOTE for an User that should have ADMIN privileges in the Guacamole Web Proxy server check ALL the "privilege" boxes as shown in the above Picture #4.   This User is being given full Admin Permissions for the Guacamole Web Proxy

The only option "normal" user's should be given is the "change password" option.

Later while logged in one of the Desktops (host, cn1 or cn2) using your browser, you can press the LEFT-side **<CTRL> <ALT> <SHIFT>** keys and a slide-out will pop up for you.

NOTE: That the LEFT-side <CTRL> <ALT> <SHIFT> keys are also the Key combination used in Guacamole to support CUT & PASTE by ALL users.
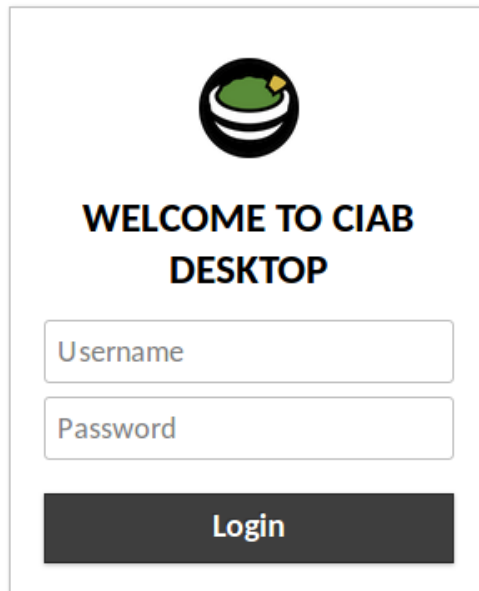
That slide-out menu will let you select to do Settings/Admin work w/out having to log-out and back in as "guacadmin".

You also must create User Accounts on the target server/Host server and in each LXD container.

The installation scripts will have already created a User Acct for yourself (as the Installer) on the Host and in both CN1 and CN2.
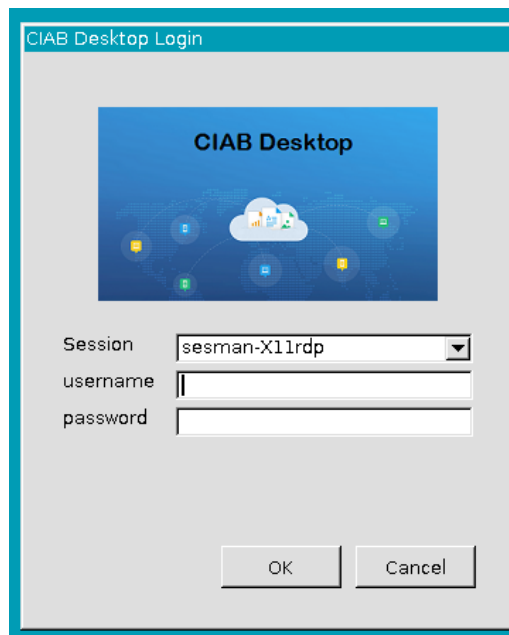
But if you need other User Accts in those containers you can do that later when you login to those containers after setup of the CIAB Remote Desktop. You will have already been give SUDO access IN those containers.

When you are doing configuration/setup of CIAB Remote Desktop in the admin screen
Click on the upper right hand corner icon (UserID will say guacadmin) and select
Log-out which will re-present the CIAB Remote Desktop login screen so you can begin
using CIAB Remote Desktop.

Picture:Remote Desktop Web Proxy Login Screen

Enter a valid login ID and password & that User will be presented with his/her own
Connection screen.   Once a User clicks on one of the Connection Icons they will
then see the XRDP Login screen where they will need to login again but this time
using their Linux UserID & Password.

Picture: Remote Desktop XRDP login menu

Again the login/password *"may"* be the same or different its up to you the installer and security policies on the target "server" or one of the 2 LXD containers.

POST INSTALLATION - ERRATA

When accessing CIAB Remote Desktop (the first time only) 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.
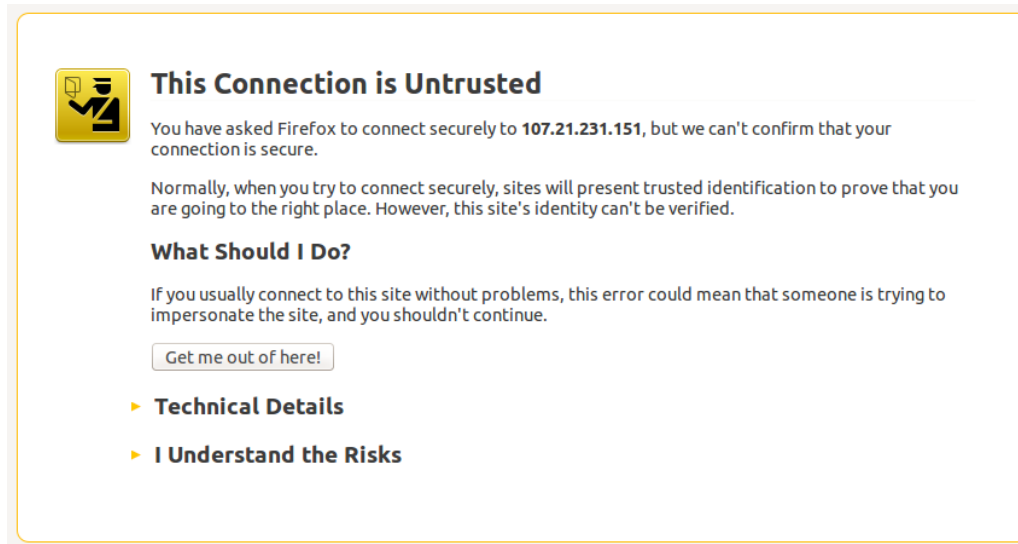
Picture: Example Web Browser warning about Unsigned Security Certificate

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.

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.

Each user will only see the above message one time (the 1st time they try to log into CIAB Remote Desktop).

During Installation, when running several of these scripts you may see a error messages various displayed concerning components of XRDP:

[....] Starting xrdp (via systemctl): xrdp.serviceA dependency job for xrdp.service failed. See 'journalctl -xe' for details.

> *Note*: You can ignore those error messages. They are caused by xrdp installation not being compatible with SystemD yet. The CIAB installation scripts take care of that for you and change things so XRDP is started with the traditional Upstart instead of SystemD.

You may also see some other errors during various apt-get installs.

These errors will be associated with "*systemd*", "*cpufrequtils*", "accountservices", "anacron" and "*dbus*".

Those also can be ignored as again they will NOT affect CIAB Remote Desktop from functioning properly.

Printing

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

Guacamole does support remote printing and you the installer can read more about it on their Wiki page:

http://guacamole.incubator.apache.org/doc/gug/users-guide.html

To enable remote printing (print from the remote Server/Host or the containers CN1 and CN2 please refer to the Ubuntu Guide to Remote Printing:

https://help.ubuntu.com/community/NetworkPrintingWithUbuntu

You might also investigate utilization of something like Google Cloud Print (requires use of Chrome).

Audio/Sound

The installation scripts hopefully have configured & enabled remote sound/audio from the Server/Host and the containers CN1 and CN2.

This is accomplished using PulseAudio's networking capabilities.

However, if you want to configure the remote audio/sound please read about how the PulseAudio module TCP is utilized/configured:

module-native-protocol-tcp

and the documentation in the Ubuntu Man Page for PACTL.

Another useful reference is Configuring LXD for audio/sound support.

As mentioned previously, the installation scripts have made an attempt to cover all of that (check the end of the host's "/etc/pulse/default.pa" file for the pulseaudio config (its basically a 1 liner utilized by the LXD containers).

Also, review the tail-end of the host's "/etc/bash.bashrc" file which provides every LXD container user with a required environment variable "PULSE_SERVER=10.x.x.x" so that while in the Container Desktop environment Pulseaudio will be redirected to the Host's Pulseaudio server.

In reality setup & configuration for the above pulseaudio/sound support is really pretty simple but it still needs more investigation/work to make it goof proof for all users of the LXD containers.

<u>**IMPORTANT USAGE NOTE IF UTILIIZING THE LXD CONTAINER DESKTOP CAPABILITY**</u>

IF you did decide to install/use LXD container desktops be aware that for the audio to work from them the User will been to login into the Host Remote Desktop first and leave that connection active before logging into either or both the the Containers CN1 and CN2.

If a user logs into one of the Containers first the Remote Desktop still works fine but there will be no audio and later if the user also logs into the Host Remote Desktop it also will not have any audio until the server itself is remoted.   So again, if audio is important and you want to use the LXD container desktop capability with audio support… log into the Host desktop 1st then log into one or both of the containers…!!

Adding more Container Remote Desktop servers

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.

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

This will create an exact copy of the CN1 container & name it CN3:

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

> **$ lxc stop cn1** (or cn2)

next clone/copy that container to a new container:

> *$ lxc copy c1 c3*

The above command would clone our CN1 container to a new container named CN3!

Restart the containers with the command:

> **$ lxc start cn1** (or cn3)

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**

NOTE:

You will need to also, manually change the Hostname (in file /etc/hostname) in the CN3 container so it doesn't keep "cn1" as its hostname.

You will also need to go back and add a new "connection" in the Guacamole Web Proxy configuration manager for the new Container CN3!

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 60 seconds or less to complete.

You could repeat the above to create any number of new containers based off of an original "base" 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 Command Line Documentation on Github](#) to learn more!

The installation scripts should set things up so that any future new users added to the Host or either of the 2 containers will be setup to receive Audio (via PulseAudio).

If Sound/Audio is not heard while logged into Host or a Container Desktop

Check #1

If for some reason Audio is not heard by a user the first thing to check is that EACH user is a member of 3 Groups (audio, pulse, pulse-access).

If a user is NOT… then issue the following command then have the user logout & log back in (userID = the user ID of the user having the problem):

> **$ sudo adduser userID pulse-access**
> **$ sudo adduser userID pulse**
> **$ sudo adduser userID audo**

The above should be executed in whichever Desktop Environment he/she doesn't get sound "from"… (host, cn1 or cn2).

However, as mentioned earlier I still have not gotten the Pulseaudio to reliably work from the LXD containers !!

Check #2 (for CN1 or CN2 Desktop Users ONLY)

For every userID you create in one of the containers CN1 or CN2 they ***MUST*** have a **PULSE_SERVER** environment variable defined that points to the IP address of the HOST machine where the PulseAudio Server daemon is running.

When you installed LXD you were prompted to create a Network for LXD with a screen looking something like the following:

```
Package configuration




                          ┤ Configuring lxd ├
   ┌──────────────────────────────────────────────────────────────────────┐
   │ Containers need a bridge to connect them together and to the host for  │
   │ outside network connectivity.                                          │
   │                                                                        │
   │ Choosing this option will let you configure the default LXD bridge to  │
   │ your liking.                                                           │
   │                                                                        │
   │ If you would rather not have LXD do this for you, then you will be      │
   │ asked whether you want to use an existing bridge or just do everything  │
   │ manually.                                                              │
   │                                                                        │
   │ Would you like to setup a network bridge for LXD containers now?        │
   │                                                                        │
   │           <Yes>                              <No>                       │
   └──────────────────────────────────────────────────────────────────────┘
```

You should have answered YES on that form which would have then presented several other forms for you to answer questions on.

The 5ᵗʰ screen/form you were presented would have looked something like the following but with some random 10.x.x.x network address listed.

For demonstration purposes I recommend changing that IP address 10.0.3.1 to become the IP address of the LXDBR0 bridge (and thus the IP address of the HOST from any container's perspective)…

NOTE: Several subsequent screens related to this will also need to be changed from whatever 10.x.x.Y address they present to 10.0.3.Y (Y = the number on the screen that you should leave set to what it already is … just change the 2 middle numbers to 0.3)

```
┤ Configuring lxd ├
A valid IPv4 address. (e.g. 10.0.8.1)

IPv4 address:

10.0.3.1

                    <Ok>
```
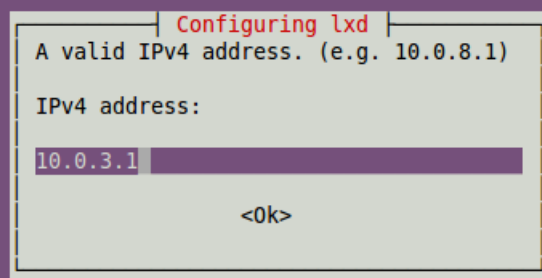
Chromium (or Chrome) Browser

In both Chrome & the Chromium Browser's there is 1 setting that **_MUST_** be changed to eliminate the possibility of a problem with remote video.

In either browser, click on its Customize & Control "button" (usually on the upper right hand of the browser.

Click on **"Settings"**
Click on **"Show Advanced Settings"**
UNCHECK the **"Use Hardware acceleration when available"** option.

Restart the browser.

     NOTE: You MUST do this in the Host & both Containers (CN1 & CN2)!

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.

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

On AWS EC2 I using Canonical's Ubuntu 16.04 Xenial Cloud Image AMI # [ami-64140d0e](#). Using that AMI I spun up what Amazon calls an **cx3.*4xlarge "instance".***

*The **cx3.4xlarge** vm provides:*

1. **16 VCPU (virtual cpu)**
2. **30 GB of memory**
3. **2 160 GB SSDs**

   NOTE: you can choose a smaller AMI/Instance type especially if you do NOT plan on using the LXD/LXC container portion of this installation Guide.

This is a large/fast cpu instance I chose because:

its very fast at installing CIAB
it would be quite sufficient if used in eventual production with possibly many LXD containers.

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 install-all.tar.gz 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 <u>entire process took 45-50 minutes</u> 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 CN2).

After the AWS Server came back online I used Chromium (re Chrome) & HTTPS to access the CIAB Remote Desktop Web Proxy on the AWS Server, logged in as "guacadmin" and used the password created when I ran setup-guacamole.sh at 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 or either CN1 or CN2 all through a Chromium browser.

Post Installation Checkbox/Checklist

(Print this off & check that you didn't forget any steps)

☐ Installed Ubuntu 16.04 server onto some Host.   That Host can be a local KVM VM or a Cloud Server on AWS or Digital Ocean.   We will call this the "Target Server"

☐ Created a directory /opt/ciab

☐ Make your UserID the owner of /opt/ciab (sudo chown userID:userID /opt/ciab

☐ Copy the install-all.tar.gz to the Target Server /opt/ciab directory

☐ Uncompress the install-all.tar.gz in /opt/ciab (cd /opt/ciab then.. sudo tar -xvf ./*.gz)

☐ Execute "$ setup-guacamole-sh" script.

   During this you will enter 3 passwords.

   One for MySql, one for NGINX and one for the Guacamole Proxy agent.   DO NOT make them all the same!

☐ Execute the "$ sudo ./setup-nginx.sh" script

☐ Reboot the VM or Cloud Server then log back in and change back to the /opt/ciab directory again

☐ Execute the "$ sudo ./setup-ciab.sh" script

☐ Execute the "$ ./setup-lxd.sh" script (run it as non-sudo user)

☐ Execute the "$ ./setup-containers.sh" script (run it as non-sudo user)

   When prompted answer questions related to LXD file system & LXD networking such as the IP address of the LXDBR0 bridge, DHCP etc.

☐ Would you like to setup a network bridge for LXD containers now… select YES

☐ Bridge Interface name: Select OK for the default name "lxdbr0"

☐ Do you want to setup an IPv4 subnet?   Select YES

☐ On next screen Select – OK

   ◦ I've configured the installation scripts to "do the right thing" and later config steps "should" detect the IPv4 address chosen by the LXD installer.

   ◦ NOTE: that will also become the IP address of the LXDBR0 bridge from inside the Containers CN1 and CN2

☐ For IPv4 CIDR mask – Select OK for the default presented

☐ Next screen select OK

☐ For Last DHCP address – Select OK

☐ For Max number of DHCP clients – Select OK

☐ Do you want to NAT the IPv4 traffic – Select YES

☐ Do you want to setup an IPv6 subnet – Select – NO (Recommend NOT to configure IPv6 at this time. You can go back and redo this later if you want/need IPv6 support for the LXD containers).

☐ During the execution of "setup-containers.sh" it will create a new User Account (userID) in the container for the UserID you are currently using.

This is exactly like creating a new user in Ubuntu.

You will be prompted for the Password for your User Account in the Container, asked to confirm it, asked for the User Name etc… just answer all those questions as normal so you can a login to the container later.

NOTE: as with Ubuntu this First User Account in the container (re You) will have SUDO privileges IN the Container (this is **NOT** the same Sudo as in the Host Server !!!).

The setup-containers.sh script will then install ubuntu-mate desktop into the Container CN1

☐ The "setup-containers.sh" script after finishing the CN1 container will copy/clone it to create the CN2 container.

☐ After the CN2 container is created the "setup-containers.sh" script will cause the display of the IP address of the CN1 and CN2 containers (fyi - the cli command is: "lxc list")

☐ Write down the IP addresses of the HOST and the CN1 and CN2 containers

☐ Reboot the HOST Server again

☐ On a different machine use Chromium (or Chrome) and point it to "https://ip_address_of_host/guacamole/guacamole"

☐ When presented with the Warning Your Connection is Not Private message screen (this is because we're using a non-valid Certificate for the Web Server click on link at the bottom labeled "ADVANCED".   Then click on the link labeled something like: Proceed to X.X.X.X (unsafe).

NOTE: You can always edit the NGINX config later and insert your own valid CERTIFICATE information to avoid this in the future.

☐ Login to Guacamole as "guacadmin" and the Password you entered previously when you installed Guacamole.

☐ In the upper right corner click on "guacadmin" in the upper right corner and select SETTINGS

☐ Create 3 new Connections by clicking on the Connections button.   One Connection for the HOST Server, one for the CN1 Container and one for the CN2 Container

☐ As you create each new "Connection" change the Connection PROTOCOL from VNC to RDP

☐ For each appropriate Connection configuration input the IP address of that destination.

For the HOST Connection enter "localhost" but when you configure each Container (CN1 & CN2) Connections use their 10.x.x.x IP address that you wrote down previously.

☐ In each Connection PROTOCOL make the Port 3389 (3389 = rdp port)

☐ In the SECURITY MODE for each Connection select RDP encryption

- ☐ Change the KEYBOARD LAYOUT to the language you use

- ☐ In the COLOR DEPTH list for each Connection select 24 BIT (32 did not work for me)

- ☐ Save each Connection as you finish each one's configuration

- ☐ At the top of the Guacamole Configuration screen click on USERS then add a new Guacamole Proxy UserID for yourself.   **NOTE:** this ID and password CAN be different from your UserID and password on Ubuntu or in any of the Containers

- ☐ Enter the Guacamole Proxy UserID

- ☐ Enter the Password for that Guacamole Proxy UserID

- ☐ Change the Time Zone appropriately to match your Location

- ☐ Check ALL boxes for PERMISSIONS. So you (the installer) can be a Guacamole Admin

- ☐ Check ALL boxes for Connections (this is just for you the Admin) other users may have only 1 or more of those boxes selected which will give them access to only those Connections you've enabled.

- ☐ Click Save to save your Guacamole Proxy User Account ID.

- ☐ Click on the "guacadmin" in the upper right corner and select LOGOUT

- ☐ Verify that you can log back in using YOUR new Guacamole Proxy UserID and Password.

- ☐ Click on you UserID in the upper Right corner and select SETTINGS again

- ☐ Verify that you now see the same Setup page as Guacadmin. If you do... then you now have Guacadmin privileges.

- ☐ Click on USERS

- ☐ Click on the UserID "Guacadmin"

- ☐ Click on DELETE to delete the Guacadmin account as its no longer needed.

- ☐ Click on your UserID again in the Upper Right corner and select HOME

- ☐ Enter your UserID and Password and you should be presented with the 3 Connections to choose from (Host, CN1 and CN2).

- ☐ Click on any one of those Connections and you will be prompted for the actual Login for that Connection (ie your Ubuntu UserID and Password that you configured).

- ☐ Enter your UserID and Password and verify that the Ubuntu-Mate desktop is presented to you.

Thats 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 (and in the CN1 or CN2 containers IF you are using LXD containers too) you will need to subsequently create a new UserID in Guacamole Web Proxy Account for that User user as well.   For each Guacamole User account created remember to check the box in the Guacamole Proxy User configuration screen for <u>each</u> "Connection" you want each Guacamole Proxy User to have access to (can be one or more of the available Connections)!*

FINAL NOTE:

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

The will be some anomalies as it was all a learning process for myself too.

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 experimental use of LXD containers could eventually enable Guacamole to support a large number of LXD container desktop targets running off a single (or cluster) powerful server (AWS cloud servers can be up to 32 core?).

Areas that still need focus are simpified Printing support/setup (this is more a Guacamole related issue) and Pulseaudio/sound hardening (this is more a Pulseaudio/LXD related issue).

Thanks for checking this out… Brian Mullan (bmullan.mail@gmail.com)