Welcome to CIAB Remote Desktop v1.0

Installation Guide

for Ubuntu 18.04 LTS

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

9/7/2018

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

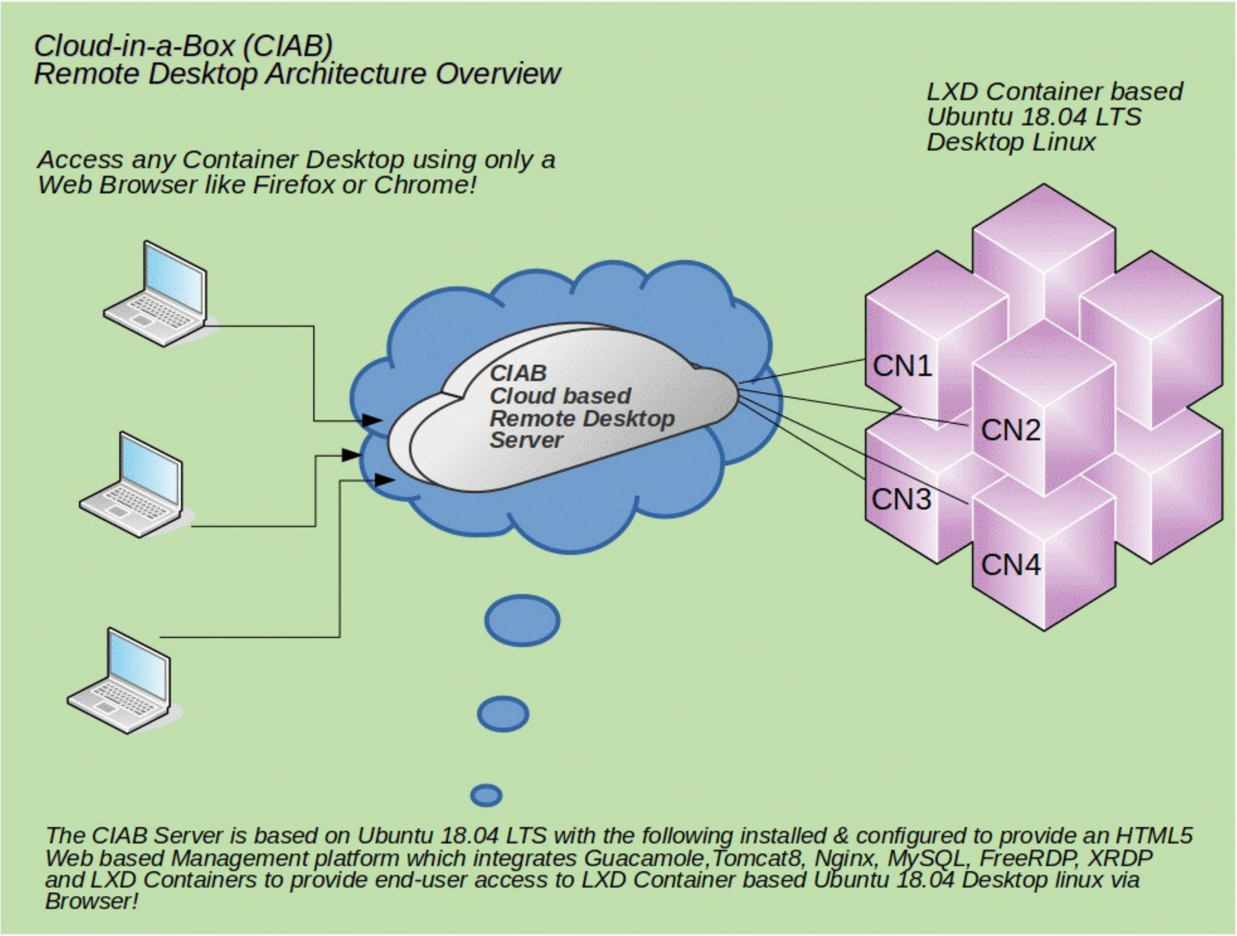
librarian, teacher or volunteers

- 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!


Cloud-in-a-Box (CIAB)
Remote Desktop Architecture Overview
LXD Container based
Ubuntu 18.04 LTS
Desktop Linux
Access any Container Desktop using only a
Web Browser like Firefox or Chrome!
CIAB
Cloud based
Remote Desktop
Server
CN1
CN2
CN3
CN4
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 the largest Virtual Machine you can spin up today approximates this:

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, 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… 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 can be 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 different use-cases.

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 6 scripts to completely automate installation of the CIAB Remote Desktop for you onto an Ubuntu 18.04 server (local/cloud or a VM).

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. 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.   These seem to perform some what better in regards to remote viewing of video/audio over the HTML5 connection.

This CIAB Remote Desktop installation process takes approximately 30-60 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.

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 ssh session.*

To begin installation unarchive 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.

```
$ sudo chmod +x /opt/ciab/*.sh
```

Then start the installation

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

During installation, two LXD containers will be created.   First, a container named CN1 is configured as a RDP enabled Remote Ubuntu MATE Desktop.

CN1 is then used to clone/copy another LXD container named ciab-guac.

Guacamole will be installed in the ciab-guac Container and you will only have to login via a Browser to Guacamole and configure Users and Connections. 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 the Guacamole Admin will require a Guacamole LoginID & Password and also a LoginID and Password on each “connection” Desktop server any User is configured/allowed to access.

The LXD containers will appear & act like a separate servers even though they run on the same Server/Host.   You could, as admin, install different applications 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.

This process will install on the Server/Host:

- CIAB Remote Desktop HTML5 web proxy is based on the great Guacamole project (see http://guacamole.apache.org/) to enable connections using an HTML5 compatible browser
- the Ubuntu-MATE desktop environment
- mysql
- nginx
- tomcat8
- xrdp & x11xrdp
- both ciab-guac and CN1 containers will have the Ubuntu-MATE desktop environment installed
- in CN1 .. 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 using either Guacamole or login just using ssh.

Note: the only place the RDP protocol is utilized is from the CIAB Remote Desktop Web Proxy running in ciab-guac LXD container and the Remote Desktop connection to the ciab-guac or CN1.

As part of the installation process all HTTPS traffic received by the Server/Host is proxy’d (forwarded) to the ciab-guac LXD container.

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 (TLS) encrypted communication capable because of our configuration of Nginx BUT … it is the responsibility of the installer to install/configure a LetsEncrypt Certificate for the HTTPS/TLS encryption to be functional!

CIAB Remote Desktop 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

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 *
# make the bash scripts executable
$ chmod +x ./*.sh
```

STEP 3

Start the installation:

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

Note: you will be prompted to input 3 passwords during this step.

- A password for MySQL root password
- A password for access to configure/change the Guacamole Database

STEP 4

Execute the script:

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

This script will create 2 containers (ciab-guac and cn1), install guacamole/mysql/tomcat/ngnx/xrdp/ubuntu-mate into ciab-guac and ubuntu-mate/xrdp in cn1.

Note: this step “may” take 20-30 minutes depending on how fast your "server" is (re does it have SSD, multiple CPU core 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 your new Remote Desktop(s) out

Note: The reboot it can take up to 2-3 minutes because of all the web servers etc that need to load & initialize (especially on a Cloud service like AWS or Digital Ocean or Hetzner).

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 and login Password

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

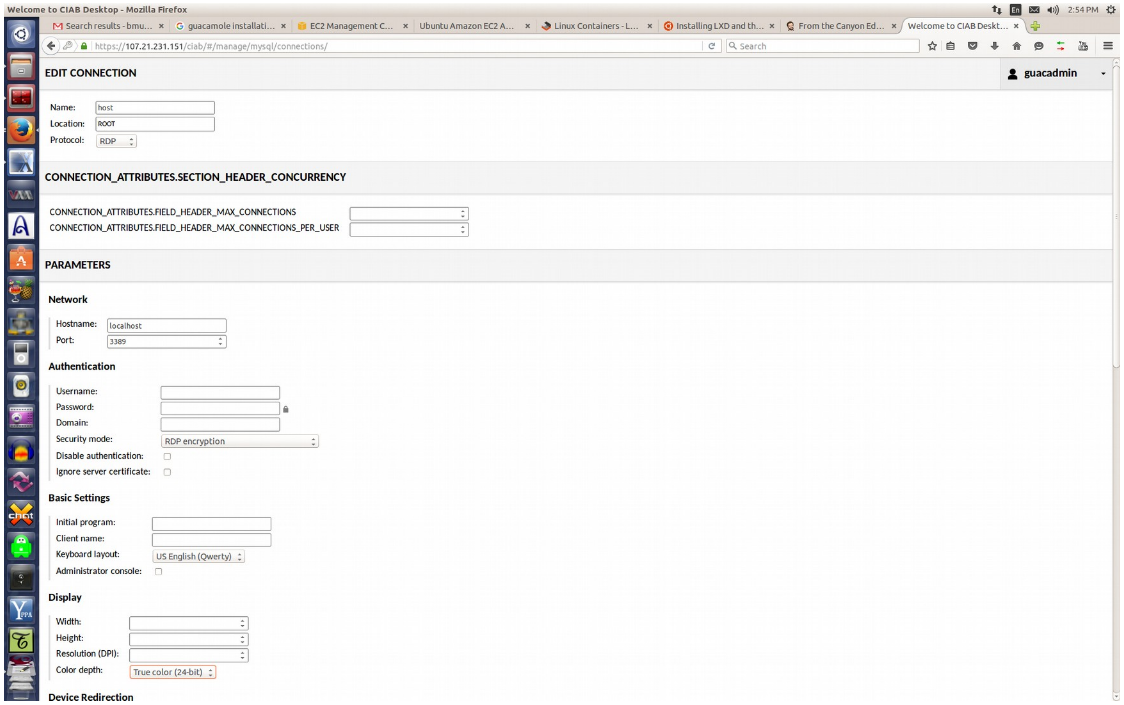
- change the Display Language to what suits you

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 etc.

Click on CONNECTIONS

- add a new connection



https://107.21.231.151/ciab/#/manage/mysql/connections/

EDIT CONNECTION

Name: host
Location: ROOT
Protocol: RDP

CONNECTION_ATTRIBUTES.SECTION_HEADER_CONCURRENCY

CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS
CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS_PER_USER

PARAMETERS

Network

Hostname: localhost
Port: 3389

Authentication

Username:
Password:
Domain:
Security mode: RDP encryption
Disable authentication: □
Ignore server certificate: □

Basic Settings

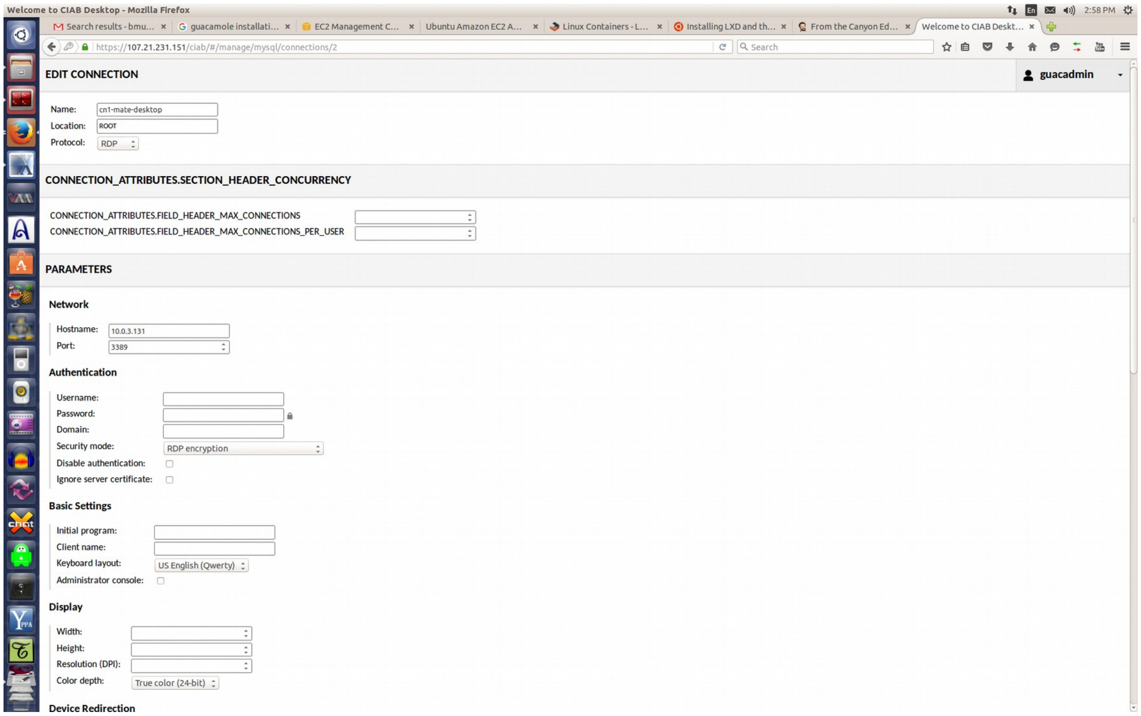
Initial program:
Client name:
Keyboard layout: US English (Qwerty)
Administrator console: □

Display

Width:
Height:
Resolution (DPI):
Color depth: True color (24-bit)

Device Redirection

Picture #1: Example of HOST Connection configuration



Welcome to CIAB Desktop - Mozilla Firefox

https://107.21.231.151/ciab/#/manage/mysql/connections/2

EDIT CONNECTION

Name: cn1-mate-desktop
Location: ROOT
Protocol: RDP

CONNECTION_ATTRIBUTES.SECTION_HEADER_CONCURRENCY

CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS
CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS_PER_USER

PARAMETERS

Network

Hostname: 10.0.3.131
Port: 3389

Authentication

Username:
Password:
Domain:
Security mode: RDP encryption
Disable authentication: □
Ignore server certificate: □

Basic Settings

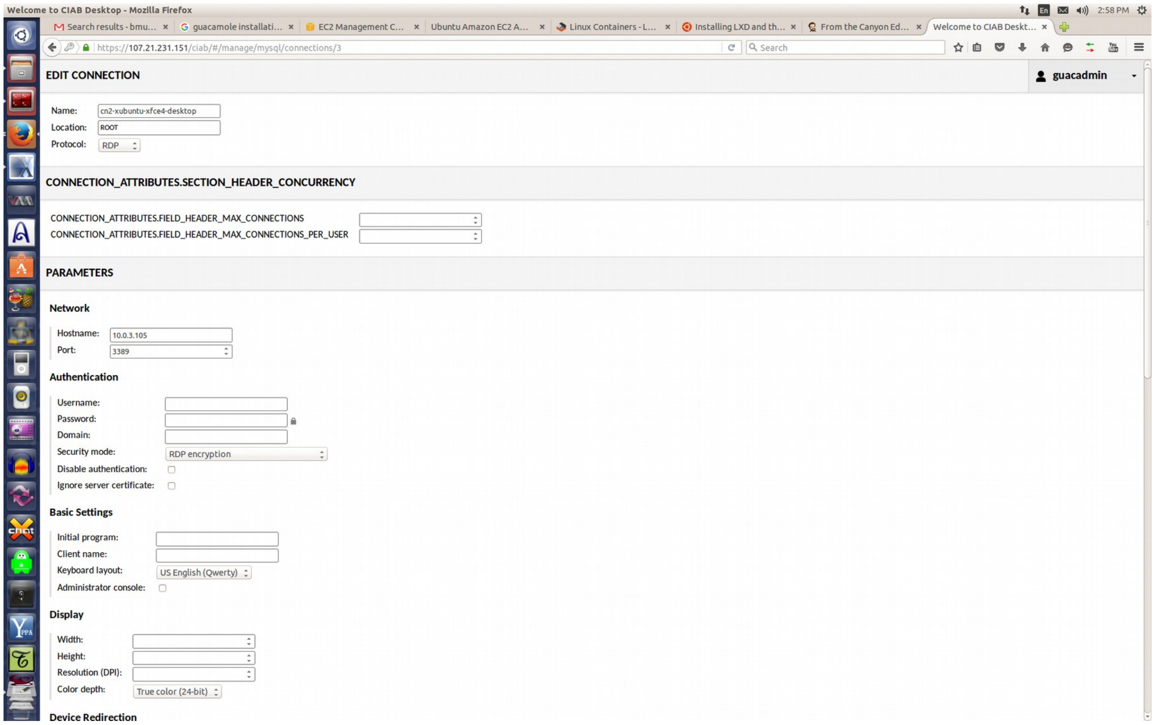
Initial program:
Client name:
Keyboard layout: US English (Qwerty)
Administrator console: □

Display

Width:
Height:
Resolution (DPI):
Color depth: True color (24-bit)

Device Redirection

Picture #2: Example of CN1 Connection configuration



Welcome to CIAB Desktop - Mozilla Firefox

https://107.21.231.151/ciab/#/manage/mysql/connections/3

EDIT CONNECTION

Name: cn2-xubuntu-xfce4-desktop
Location: ROOT
Protocol: RDP

CONNECTION_ATTRIBUTES.SECTION_HEADER_CONCURRENCY

CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS
CONNECTION_ATTRIBUTES.FIELD_HEADER_MAX_CONNECTIONS_PER_USER

PARAMETERS

Network

Hostname: 10.0.3.105
Port: 3389

Authentication

Username:
Password:
Domain:
Security mode: RDP encryption
Disable authentication: □
Ignore server certificate: □

Basic Settings

Initial program:
Client name:
Keyboard layout: US English (Qwerty)
Administrator console: □

Display

Width:
Height:
Resolution (DPI):
Color depth: True color (24-bit)

Device Redirection

Picture #3: Example of CN2 Connection configuration

Repeat Step 2 for the LXD container 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” 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 connection – enter the IP address you wrote down that was displayed during installation of the CN1 container. They will 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** or 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 LXD container !!

These IDs are only used to allow access to the CIAB Remote Desktop web proxy system using an HTML5 compatible Browser.

Remember for EACH user you create in Guacamole 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.

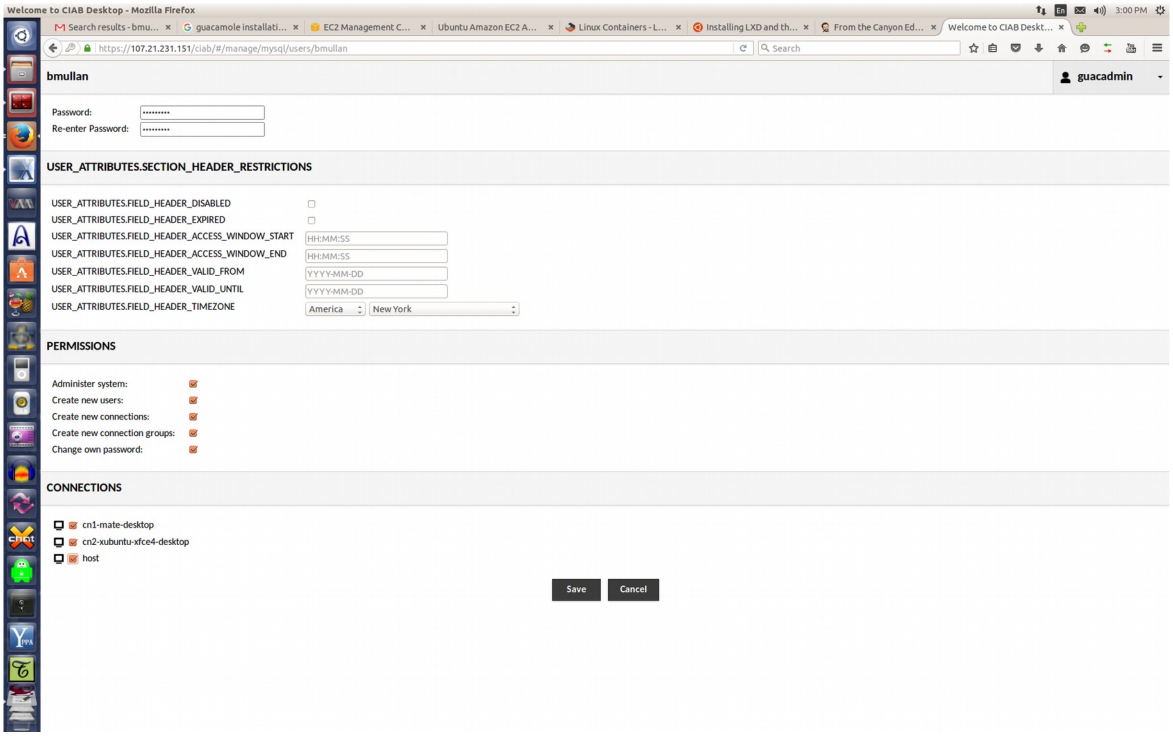
TIP: you may want to check the box to let them change their own password! After successful login/password the users will get a "***Connections***" menu (configured by the admin) where they can click on any Connection, you as the Admin, have enabled for them.

Note: *you can* make the Login ID and Password in the Guacamole Web Proxy *different or the same* as the Linux User's ID and pwd you created on the Host or in the container CN1.

First, create a Guacamole UserID and Password for yourself (the installer)!

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

NOTE: After saving your User information you can logoff Guacamole as the GuacAdmin user and log back in as you UserID and then delete the GuacAdmin UserID or change that UserID's password to something better than the Guacamole installation default of "guacadmin"! If you keep the GuacAdmin UserID do NOT leave its password as "guacadmin"… change it.



Welcome to CIAB Desktop - Mozilla Firefox

https://107.21.231.151/ciab/#/manage/mysql/users/bmullan

bmullan

guacadmin

Password: ••••••••••

Re-enter Password: ••••••••••

USER_ATTRIBUTES.SECTION_HEADER_RESTRICTIONS

USER_ATTRIBUTES.FIELD_HEADER_DISABLED ☐

USER_ATTRIBUTES.FIELD_HEADER_EXPIRED ☐

USER_ATTRIBUTES.FIELD_HEADER_ACCESS_WINDOW_START HH:MM:SS

USER_ATTRIBUTES.FIELD_HEADER_ACCESS_WINDOW_END HH:MM:SS

USER_ATTRIBUTES.FIELD_HEADER_VALID_FROM YYYY-MM-DD

USER_ATTRIBUTES.FIELD_HEADER_VALID_UNTIL YYYY-MM-DD

USER_ATTRIBUTES.FIELD_HEADER_TIMEZONE America New York

PERMISSIONS

Administer system: ☑

Create new users: ☑

Create new connections: ☑

Create new connection groups: ☑

Change own password: ☑

CONNECTIONS

☑ cn1-mate-desktop

☑ cn2-xubuntu-xfce4-desktop

☑ host

Picture #4:   Adding a new User configuration.

NOTE for a 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 unless they are a "trusted" user.

Later while logged in one of the Desktops (the Host or CN1) 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.

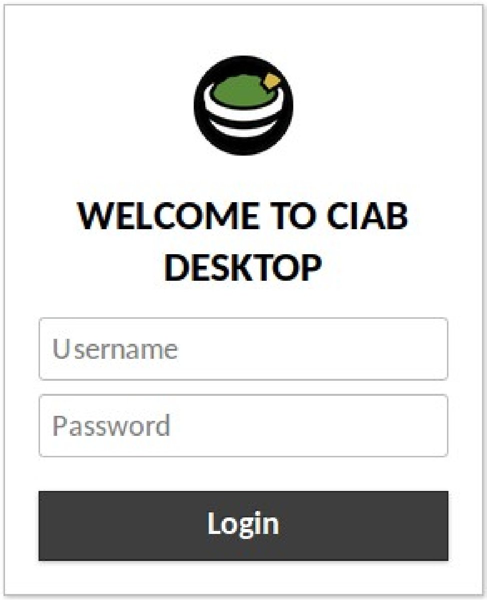
That slide-out menu will let you select to do Settings/Admin work w/out having to log-out and back in as "guacadmin".

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.

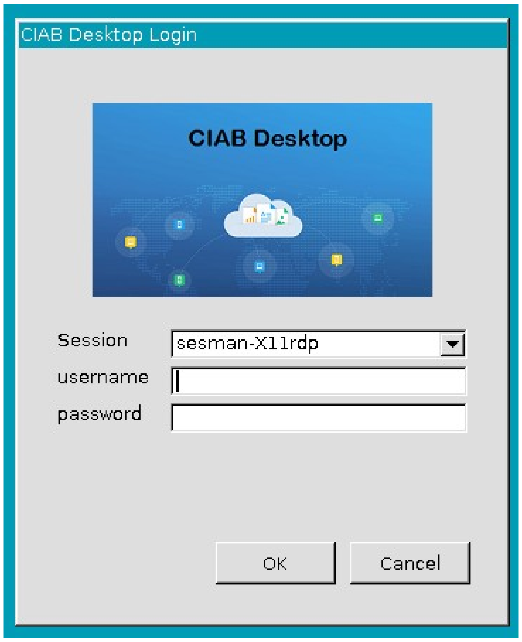
When you are doing configuration/setup of CIAB Remote Desktop in the admin screen Click on the upper right hand corner icon (UserID will say guacadmin) and select Log-out which will re-present the CIAB Remote Desktop login screen so you can begin using CIAB Remote Desktop.



WELCOME TO CIAB 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 for whichever “connection” (ie Desktop Server – the Host or CN1) they want to access & use.



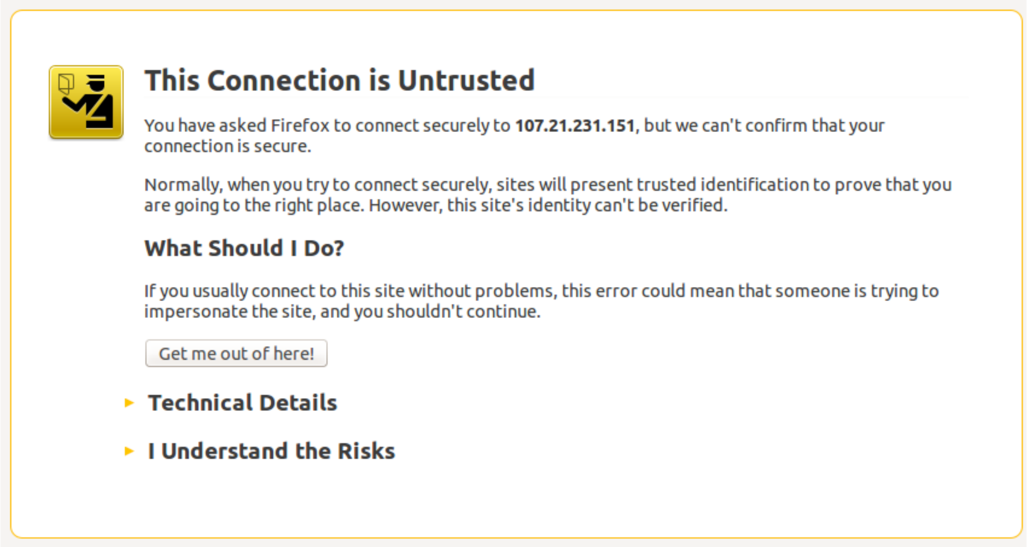
Session sesman-X11rdp
username
password

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.



This Connection is Untrusted

You have asked Firefox to connect securely to **107.21.231.151**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

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).

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 container CN1 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 container CN1.

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 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
```

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!

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 will 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 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). Note that the installation scripts should set things up so any newly created User Accounts will automatically be added to those 3 “groups”.

Chromium (or Chrome) Browser

In both Chrome & the Chromium Browser's there is 1 setting that is recommended to be changed to eliminate the possibility of a problem with remote video/audio. If you have any problems with remote desktop video/audio then try the following

In your browser, click on its customize & control “button” (3 vertical Dots usually on the upper right hand corner 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 & the Container CN1!

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 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 20-30 minutes</u> (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 contaiers).

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.

Post Installation Checkbox/Checklist

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

□ Installed Ubuntu 18.04 server onto some Host. That Host can be a local KVM VM or a Cloud Server on AWS , Digital Ocean Hetzner etc. 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-ciab.sh" script.

During this you will be asked several times to enter passwords, one for MySql, one for the Guacamole Database access.

When prompted answer questions related to LXD file system & LXD networking such as the IP address of the LXDBR0 bridge, DHCP etc. All options for default values could be accespted except for the following 3.

□ 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 ciab-guac and CN1

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

□ Next screen select OK

□ 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).

□ Write down the IP addresses of the ciab-guac and the CN1 container

□ Reboot the HOST Server again

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

□ When presented with the about "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 & insert a valid Certificate (LetsEncrypt is a popular and free source of valid Certificates) to avoid this in the future.

□ Login to Guacamole as "guacadmin" and the password "guacadmin".

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

□ Create 2 new Connections by clicking on the Connections button. One Connection for the HOST Server, one for the CN1 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 ciab-guac Connection enter "localhost" but when you configure the CN1 Container Connection use its 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 (or 32 Bit)

□ 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

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 UserID for the new user.
- □ Enter the Password for that Guacamole 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 2 Connections to choose from (Host or CN1).
- □ 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.

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 Guacamole for that User user as well. For each 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 NOTE:

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 v1.0 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-guac and cn1 containers to that Repository. Then for future new installations all you'd have to do is install a server somewhere, install LXD on it and then use the LXD CLI to copy and create a new ciab-guac and a new CN1 container. That should only take 3-5 minutes as you already have those containers working and they are now just a "based" for launching more.

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)