Install Guacamole using Docker

https://www.youtube.com/embed/DGw6P5Lkj-U

A really terrific RDP client that makes your remote machine desktops and environments available through your web browser. The docker image we'll be using was built off of the original Oznu image work, but has been updated to provide some of the more recent features.

JWetzell's image allows for the extended authentication extensions to be used, and this really adds a lot of great security to the offering. The docker-compose.yml is very straight-forward and clean.

Installation

What you'll need

- Docker-CE and Docker-Compose installed
- (optional) Portainer-CE
- (optional) NGinX Proxy Manager
- (optional) Authelia (for an additional security layer)
- About 10 minutes of your time.

Installing Docker-CE and Docker-Compose

You can easily install Docker-CE, Docker-Compose, Portainer-CE, and NGinX Proxy manager by using this quick install script I created and maintain on Github. Just use the command:

```
wget https://gitlab.com/bmcgonag/docker_installs/-/raw/main/install_docker_nproxyman.sh
```

To download the script to your desired host.

Change the permissions to make the script executable:

```
chmod +x ./install_docker_nproxyman.sh
```

and then run the script with the command:

```
./install_docker_nproxyman.sh
```

When run, the script will prompt you to select your host operating system, then will ask you which bits of software you want to install.

Simply enter 'y' for each thing you want to install.

At some point, you may be asked for your super user (sudo) password as well.

Allow the script to complete installation.

At this point, you might want to log out and back in, as this will allow you to use the `docker` and `docker-compose` commands without the need of sudo in front of them.

Installing Guacamole with Docker and Docker-Compose

Let's create a good base structure for keeping our docker organized, and easy to backup and restore if needed. We'll make a folder called "docker" that we'll keep everything in.

`mkdir docker`

Next, let's move into that folder, and continue working from there.

`cd docker`

Now, we'll make a new folder called "guacamole":

`mkdir guacamole`

Let's move into our new guacamole folder, and then we'll create a new file in that directory called "docker-compose.yml":

`cd guacamole`

`nano docker-compose.yml`

Inside that file we just opened, we want to copy and paste the code block below.

```
version: "3"
services:
  guacamole:
    image: jwetzell/guacamole
    container_name: guacamole
    volumes:
      - ./postgres:/config
    ports:
```

```
      - 8080:8080
volumes:
  postgres:
    driver: local
```

After pasting the code-block above into your nano text editor (use right-click >> paste, or CTRL + Shift +V to paste), you may want / need to adjust the port mapping to a different port. I find that port 8080 is a very common port, so I like to change it to something less common to avoid port conflicts with other containers I run.

To change a port mapping, change the port number on the left side of the colon only. Never change the right side, as the right side is what the container is expecting for the port to be. The left side, however, is the port for your host machine, and you can change that to any unused (open) port on your host.

In this case, I used port 8190 as my open port for guacamole.

so my port mapping section looks like:

```
    ports:
      - 8190:8080
```

Once you've made this change (if you want / need to), save the file with CTRL + O, then press Enter to confirm, and exit the nano text editor with CTRL + X.

We are ready to bring up our Guacamole Remote Access application.

In the terminal at a fresh prompt, simply enter the command:

`docker-compose up -d`

Allow the images to be pulled down, and the container to start. You should see a "done" message in the terminal when the process has completed.

You can use a tool like Portainer to easily check the logs. If you don't have portainer, not to worry. You can use the command

`docker-compose logs -f`

to view the logs of the container. We are simply checking for any errors in the logs that might indicate our install didn't work for some reason. Hopefully you'll find no errors, and you can now open a browser window and navigate to the IP address of your host on the port you set on the left side of the port mapping.

In my case it was `http://192.168.10.147:8190`

You should now be greeted with the guacamole login screen.

Login with the default credentials of:

- username: guacadmin
- password: guacadmin

Setting up a new Guacamole User and Deleting the Default User

In the Guacamole interface, you'll want to go to the upper right and click on the user avatar. This will present a drop-down menu of options. Select the 'Settings' option, and then you'll see a set of tabs with which you can setup the system.

The first thing I like to do is to create a new user (under the users tab), and make that user an Admin of the system by checking all of the check boxes for user permissions. I give the user a strong password, and save.

Next, I log out of the default "guacadmin" user, and login as my newly created user. Once logged in as the new user, I go back into "Settings >> Users" and I delete the "guacadmin" default user by clicking on the username for "guacadmin", scrolling to the bottom of the edit screen, and selecting the "Delete" button, then confirming.

You can now add more users if you have others who you'd like to have access to your Guacamole instance. I suggest, however, that you first add all of the Remote Connections you'll need, then add the users, as part of the User add process is selecting which machines / groups of machines each user should be able to access.

Setup Connections and Connection Groups

Think about the machines you'll be adding. If it's only for home use, and only 1 or 2 machines, you may not need to create connection Groups. If, however, you'll be adding multiple machines for home, work, clients, end-useer support, etc, then you may want to consider the best groupings of those machines, and first create some Connection Groups. This will allow you to organize your connectins in a way that makes them easier to locate later.

You might create Connection Groups like:

- Home / Personal
- Client Desktops
- Web Servers
- Document Storage Servers
- Media Servers
- End User Consoles

Really, you can group machines in so many ways, i couldn't possibly provide a decent list here. You can always add connections to a group later as well.

Adding a new Connection is as easy as selecting the Connections tab in the Settings area, and clicking 'New Connection'.

Now just select how you want to connect, and provide the necessary connection details (IP or Hostname, Port, User Credentials, etc), and then save. One really great feature is the "Clone" capability. If you're setting up a lot of machines that are essentially the same settings and just different IPs and User Credentials, it will be a real time saver to make your first Connection, save it, then open it to Edit mode, and scroll to the bottom to click the Clone button. As soon as you click clone, you are in the Cloned connection, and ready to edit it.

Support my Channel

Support my Channel and ongoing efforts through Patreon:

https://www.patreon.com/bePatron?u=234177

Revision #1
Created 2022-02-24 19:13:04 UTC by Brian McGonagill
Updated 2022-02-24 19:15:55 UTC by Brian McGonagill