Apache Guacamole Docker Deployment

After you've installed docker, you can now setup a environment for Apache Guacamole.

Step One: Deploy guacamole-server

Hosts the guacd container (apache/guacamole-server).

On 11/5/2019: The docker container provided by the apache/guacamole-server Dockerfile does not build cleanly into a docker container. This is primarily due to the Dockerhub file using base image debian:stable (buster) at this time. Debian Buster lacks the previous freerdp 1.x library files and now uses freerdp 2.0 files. Mike Jumper is building in freerdp 2.x support into guacamole v1.1.0. Progress of FreeRDP v2.0 implementation can be tracked via GUACAMOLE-249 (https://issues.apache.org/jira/browse/GUACAMOLE-249).

I've changed apache/guacamole-server Dockerfile to use Debian Stretch for the moment for testing purposes since it still allows using FreeRDP v1.x libraries. That container is available on DockerHub as mabarkdoll/guacamole-server. You'll notice I use that container instead of that advised container by Apache Guacamole.

Ideally a new release of Guacamole v1.1.0 will be comitted soon to support FreeRDP 2.0 (we can switch to Mike Jumper's branch that is testing FreeRDP 2.0 at a later point if needed https://github.com/mike-jumper/guacamole-server/tree/WIP-freerdp-migrate-2.x however it shouldnt be required at this time since we're primarily working with the apache/guacamole-client docker container for HA purposes).

Download the image from dockerhub

```
$ sudo docker pull mabarkdoll/guacamole-server
[sudo] password for admin:
Using default tag: latest
latest: Pulling from mabarkdoll/guacamole-server
af4b0a2388c6: Pull complete
dcdd9c2ece80: Pull complete
674a18e16f54: Pull complete
0e6dd708fc82: Pull complete
300b33db8291: Pull complete
Digest:
sha256:3cb2ad5cfecda2bc3b5383a272d22c106b7e860cebe0600345c57f4b61e4484d
Status: Downloaded newer image for mabarkdoll/guacamole-server:latest
```

Running the guacd Docker image

E.g.,

```
$ sudo docker run --name some-guacd -d mabarkdoll/guacamole-server
```

```
$ sudo docker ps
CONTAINER ID        IMAGE                            COMMAND
CREATED              STATUS              PORTS                NAMES
299c73a7e779        mabarkdoll/guacamole-server   "/bin/sh -c '/usr/lo…"
About a minute ago   Up About a minute   4822/tcp             some-guacd
```

Step Two: Deploy MySQL container

To use Guacamole with the MySQL authentication backend, you will need either a Docker container running the mysql image, or network access to a working installation of MySQL. The connection to MySQL can be specified using either environment variables or a Docker link.

Pull the mysql docker image from DockerHub

```
docker pull mysql
####docker pull mysql:5.7.28 #(We no longer need to use the older mysql
docker container)
```

Run the mysql server

```
docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=secret -p 3306:3306
mysql:latest
####docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=secret -p 3306:3306
mysql:5.7.28 #(Run the latest instead)
```

Initializing the MySQL database

Option 2a:

You can use the SQL scripts included with the database authentication.

Once this script is generated, you must:

1. Create a database for Guacamole within MySQL, such as guacamole_db.
2. Create a user for Guacamole within MySQL with access to this database, such as guacamole_user.
3. Run the script on the newly-created database.

The process for doing this via the mysql utility included with MySQL is documented in Chapter 6, Database authentication.

I used the following steps to initialize the MySQL Server:

Upon a shell inside the mysql container

```
sudo docker exec -it some-mysql /bin/bash
```

Manually log into the mysql server and created a database and granted access to a new user.

```
mysql -h localhost -p
# Enter password secret
```

```
CREATE DATABASE guacamole_db;
CREATE USER 'guacamole_user'@'localhost' IDENTIFIED BY 'some_password';
GRANT SELECT,INSERT,UPDATE,DELETE ON guacamole_db.* TO 'guacamole_user'@'localhost';
FLUSH PRIVILEGES;
quit
```

This following initializes the database, there might've been some conflicts with the previous steps.

```
apt-get update
apt-get install -y wget
wget https://raw.githubusercontent.com/glyptodon/guacamole-client/master/extensions/guacamole-auth-jdbc/modules/guacamole-auth-jdbc-mysql/schema/001-create-schema.sql
wget https://raw.githubusercontent.com/glyptodon/guacamole-client/master/extensions/guacamole-auth-jdbc/modules/guacamole-auth-jdbc-mysql/schema/002-create-admin-user.sql
cat *.sql | mysql -u root -p guacamole_db
# Enter password secret
```

For some reason, I had to use user root rather than guacamole_user when accessing the mysql database. This might've been some permission issue with the mysql container. I didn't spend time to figure it out, since I planned to use a production mysql server later.

```
# We're done setting up the mysql container
exit
```

Option 2b: (untested)

Alternative approach provided by the documentation:

You can alternatively try the following command when initially running the mysql server.

Your database is not already initialized with the Guacamole schema, you'll need to do so prior to using Guacamole. A convenience script for generating the necessary SQL to do this is included in the Guacamole image.

To generate a SQL script which can be used to initialize a fresh MySQL database as documented in Chapter 6, Database authentication:

```
$ docker run --rm guacamole/guacamole /opt/guacamole/bin/initdb.sh --mysql
> initdb.sql
```

Optional: Enable debugging for guacamole-client container

Create custom GUACAMOLE_HOME dir

```
mkdir -p ~/guacamole/test
```

Enable debugging:

You need to have logback.xml inside the guacamole-client container for debugging to be displayed. Here we create the file and later attach it to the docker container upon launch.

```
$ echo "\
<configuration>

    <!-- Appender for debugging -->
    <appender name="GUAC-DEBUG"
class="ch.qos.logback.core.ConsoleAppender">
        <encoder>
            <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} -
%msg%n</pattern>
        </encoder>
    </appender>

    <!-- Log at DEBUG level -->
    <root level="debug">
        <appender-ref ref="GUAC-DEBUG"/>
    </root>

</configuration>" > ~/guacamole/test/logback.xml
```

Running the apache/guacamole-client container

Option 1: Run without logback debugging.xml volume share

Preferred method of running without debugging.

```
sudo docker run --name some-guacamole --link some-guacd:guacd --link some-
mysql:mysql -e MYSQL_DATABASE=guacamole_db -e MYSQL_USER=root -e
MYSQL_PASSWORD=secret -e MYSQL_HOSTNAME=some_mysql -e MYSQL_PORT=3306 -v
/home/cisadmin/guacamole/test:/home/admin/guacamole/test -p 8080:8080
mabarkdoll/guacamole-client
```

```
# Alternatively, using guacamole's official dockerhub container (not
recommended 11/5)
# 11/5: We'll not use the following since we're making a custom guacamole-
client.
# 11/5: The guacamole/guacamole dockerhub container lacks newest mysql
server support.
# 11/5: The following command is just for documentation purposes and can be
used once dockerhub has newer containers for version 1.1.

###sudo docker run --name some-guacamole --link some-guacd:guacd --link
some-mysql:mysql -e MYSQL_DATABASE=guacamole_db -e MYSQL_USER=root -e
MYSQL_PASSWORD=secret -e MYSQL_HOSTNAME=some_mysql -e MYSQL_PORT=3306 -v
/home/cisadmin/guacamole/test:/home/admin/guacamole/test -p 8080:8080
guacamole/guacamole
```

Option 2: Run with logback debugging.xml volume share

With logback debugging, note were sharing local directory /home/cisadmin/guacamole/test as directory /home/admin/guacamole/test inside the docker container. We're also specifying the location of the GUACAMOLE_HOME directory so it knows where to find the logging.xml file `-e GUACAMOLE_HOME=/home/admin/guacamole/test`.

```
sudo docker run --name some-guacamole --link some-guacd:guacd --link some-
mysql:mysql -e MYSQL_DATABASE=guacamole_db -e MYSQL_USER=root -e
MYSQL_PASSWORD=secret -e MYSQL_HOSTNAME=some_mysql -e MYSQL_PORT=3306 -v
/home/cisadmin/guacamole/test:/home/admin/guacamole/test -e
GUACAMOLE_HOME=/home/admin/guacamole/test -p 8080:8080
mabarkdoll/guacamole-client
```

How to perform the same action using official dockerhub containers: Avoid this one as of 11/5:

```
###sudo docker run --name some-guacamole --link some-guacd:guacd --link
some-mysql:mysql -e MYSQL_DATABASE=guacamole_db -e MYSQL_USER=root -e
MYSQL_PASSWORD=secret -e MYSQL_HOSTNAME=some_mysql -e MYSQL_PORT=3306 -v
```

```
/home/cisadmin/guacamole/test:/home/admin/guacamole/test -e
GUACAMOLE_HOME=/home/admin/guacamole/test -p 8080:8080 guacamole/guacamole
```

How to remove old container

When you go to run a container you might find that one already exists and you need to deploy a newer build. Here is how to remove the previous container.

```
docker ps
docker rm de56c472613f5daaec9af0c14555e6da36afc28867e83f22f915657a8fcc5157
```

How to restart a container

```
docker ps -a
docker container restart
de56c472613f5daaec9af0c14555e6da36afc28867e83f22f915657a8fcc5157
```

Test it out

http://localhost:8080/guacamole

username: guacadmin password: guacadmin

Optionally, assume bash control of a container named some-guacamole

```
sudo docker ps
sudo docker exec -it some-guacamole /bin/bash
```

Optional Customizations

How to build a custom docker container

Compile the guacamole-client container image

Checkout a particular branch

Here we merge a branch `jira/234` into master and then build the container in the next step.

```
mkdir ldap
cd ldap
git clone https://github.com/necouchman/guacamole-client.git
cd guacamole-client
git checkout master
git pull origin jira/234
```

Compile guacamole-client container into local repository

```
sudo docker build -t mbarkdoll-test/guacamole .
```

```
Successfully built 36da894e9194
Successfully tagged mbarkdoll-test/guacamole:latest
```

Pass custom guacamole.propeties file

Some of the following ip addresses might need to be adjusted based on your docker container ip addresses that were assigned.

```
$ echo "\
guacd-hostname: 172.17.0.2
guacd-port: 4822
mysql-hostname: 172.17.0.3
mysql-port: 3306
mysql-database: guacamole_db
mysql-username: root
mysql-password: secret" > ~/guacamole/test/guacamole.properties
```