

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

Download the image from dockerhub

```
1 | $ sudo docker pull guacamole/guacd
```

Output:

```
1 | [sudo] password for admin:
2 | Using default tag: latest
3 | latest: Pulling from guacamole/guacd
4 | af4b0a2388c6: Pull complete
5 | dcdd9c2ece80: Pull complete
6 | 674a18e16f54: Pull complete
7 | 0e6dd708fc82: Pull complete
8 | 300b33db8291: Pull complete
9 | Digest:
   | sha256:3cb2ad5cfecda2bc3b5383a272d22c106b7e860cebe0600345c57f4b61e4484d
10 | Status: Downloaded newer image for guacamole/guacd:latest
```

Running the guacd Docker image

E.g.,

```
1 | $ sudo docker run --name some-guacd -d guacamole/guacd
```

Output:

```
1 | 64321f7931472acfcf4a69465f335ab234c0eaa146ab4b5ff5350c8c7b0d0d8d
```

```
1 | $ sudo docker ps
```

1	CONTAINER ID	IMAGE	COMMAND	CREATED
	STATUS	PORTS	NAMES	
2	64321f793147	guacamole/guacd	"/usr/local/sbin/gua..."	25 seconds
	ago	Up 24 seconds	4822/tcp	some-guacd
3	\$			

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

```
1 | docker pull mysql
```

Run the mysql server

```
1 | docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=secret -p 3306:3306
mysql:latest
```

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

```
1 | 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.

```
1 mysql -h localhost -p
2 CREATE DATABASE guacamole_db;
3 CREATE USER 'guacamole_user'@'localhost' IDENTIFIED BY 'some_password';
4 GRANT SELECT,INSERT,UPDATE,DELETE ON guacamole_db.* TO
  'guacamole_user'@'localhost';
5 FLUSH PRIVILEGES;
6 quit
```

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

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

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.

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:

```
1 $ docker run --rm guacamole/guacamole /opt/guacamole/bin/initdb.sh --mysql >
  initdb.sql
```

Step Three: Deploy the Guacamole Docker image

Compile the guacamole-client container image

Checkout a particular branch

```
1 mkdir ldap
2 cd ldap
3 git clone https://github.com/michaelbarkdoll/guacamole-client.git
4 cd guacamole-client
5 git checkout master
6 git pull origin jira/234
```

Compile guacamole-client

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

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

Create custom GUACAMOLE_HOME dir

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

Enable debugging:

```
1 $ echo "\
2 <configuration>
3
4     <!-- Appender for debugging -->
5     <appender name="GUAC-DEBUG"
6         class="ch.qos.logback.core.ConsoleAppender">
7         <encoder>
8             <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} -
9             %msg%n</pattern>
10         </encoder>
11     </appender>
12
13     <!-- Log at DEBUG level -->
14     <root level="debug">
15         <appender-ref ref="GUAC-DEBUG"/>
16     </root>
```

```
15
16 </configuration>" > ~/guacamole/test/logback2.xml
```

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

```
1 $ echo "\
2 guacd-hostname: 172.17.0.2
3 guacd-port: 4822
4 mysql-hostname: 172.17.0.3
5 mysql-port: 3306
6 mysql-database: guacamole_db
7 mysql-username: root
8 mysql-password: secret
9 ldap-hostname: ad.siu.edu
10 ldap-port: 636
11 ldap-encryption-method: ssl
12
13 ldap-user-base-dn: dc=ad,dc=siu,dc=edu
14 ldap-username-attribute: sAMAccountName
15 # Uncomment your line and update your dawg tag
16 #ldap-search-bind-dn: cn=Ldap Query2,ou=Users,ou=CS,ou=COS,ou=Academic
    Affairs,dc=ad,dc=siu,dc=edu
17 #ldap-search-bind-dn: cn=Mark R Beussink-
    SIU853249208,ou=Student,ou=Roles,ou=IDM,dc=ad,dc=siu,dc=edu
18 # Put your sso password in the line below
19 ldap-search-bind-password: I am the captain of my soul!
20
21 ldap-follow-referrals: false
22 ldap-user-search-filter: (objectClass=user)
23 ldap-operation-timeout: 180
24 ldap-max-search-results: 30000
25 ldap-max-referral-hops: 20" > ~/guacamole/test/guacamole.properties
```

Run the container

```
1 $ sudo docker run \  
2 --name some-guacamole \  
3 --link some-guacd:guacd \  
4 --link some-mysql:mysql \  
5 -e MYSQL_DATABASE=guacamole_db \  
6 -e MYSQL_USER=root -e MYSQL_PASSWORD=secret -e \  
7 MYSQL_HOSTNAME=some_mysql \  
8 -e MYSQL_PORT=3306 -e "LDAP_PORT=636" \  
9 -e "LDAP_ENCRYPTION_METHOD=ssl" \  
10 -e LDAP_HOSTNAME=ad.siu.edu -e \  
11 "LDAP_USER_BASE_DN=dc=ad,dc=siu,dc=edu" \  
12 -e "EXTENSIONS=auth-ldap" \  
13 -v /home/admin/guacamole/test:/home/admin/guacamole/test -e  
    GUACAMOLE_HOME=/home/admin/guacamole/test -p 8080:8080 mbeussink-  
    test/guacamole
```

Remove old container

```
1 docker rm de56c472613f5daaec9af0c14555e6da36afc28867e83f22f915657a8fcc5157
```

Test it out

<http://localhost:8080/guacamole>

username: guacadmin password: guacadmin

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

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