#Reference  
# <https://guacamole.apache.org/doc/0.9.1/gug/mysql-auth.html#:~:text=The%20default%20user%20is%20%22guacadmin,user%20in%20the%20administration%20screen>.

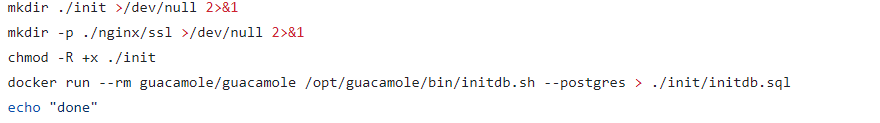
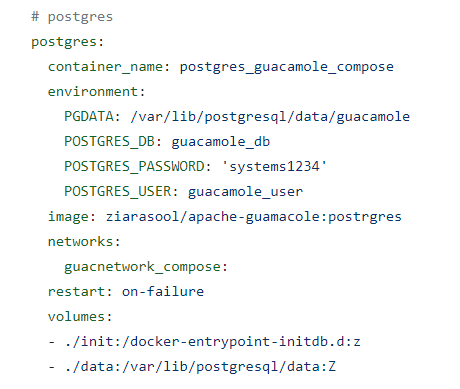
#https://www.reddit.com/r/selfhosted/comments/fnjrsg/docker\_with\_guacamole\_nginx\_struggling\_to\_reverse/

#https://www.ibm.com/docs/en/api-connect/2018.x?topic=overview-generating-self-signed-certificate-using-openssl

#https://hub.docker.com/r/guacamole/guacamole

#https://guacamole.apache.org/doc/gug/guacamole-architecture.html

To understand some details let's take a closer look at parts of the docker-compose.yml file:

* Networking
  + The following part of docker-compose.yml will create a network with name guacnetwork\_compose in mode bridged.
  + ...
  + # networks
  + # create a network 'guacnetwork\_compose' in mode 'bridged'
  + networks:
  + guacnetwork\_compose:
  + driver: bridge
* Services
  + The following part of docker-compose.yml will create the guacd service. guacd is the heart of Guacamole which dynamically loads support for remote desktop protocols (called "client plugins") and connects them to remote desktops based on instructions received from the web application. The container will be called guacd\_compose based on the docker image guacamole/guacd connected to our previously created network guacnetwork\_compose. Additionally we map the 2 local folders ./drive and ./record into the container. We can use them later to map user drives and store recordings of sessions.
  + 
* PostgreSQL
  + The following part of docker-compose.yml will create an instance of PostgreSQL using the official docker image. This image is highly configurable using environment variables. It will for example initialize a database if an initialization script is found in the folder /docker-entrypoint-initdb.d within the image. Since we map the local folder ./init inside the container as docker-entrypoint-initdb.d we can initialize the database for guacamole using our own script (./init/initdb.sql)
  + 
  + 
* Guacamole
  + The following part of docker-compose.yml will create an instance of guacamole by using the docker image guacamole from docker hub. It is also highly configurable using environment variables. In this setup it is configured to connect to the previously created postgres instance using a username and password and the database guacamole\_db. 
* nginx
  + The following part of docker-compose.yml will create an instance of nginx that maps the public port 8443 to the internal port 443. The internal port 443 is then mapped to guacamole using the ./nginx/templates/guacamole.conf.template file. The container will use the previously generated (script.sh) self-signed certificate in ./nginx/ssl/ with ./nginx/ssl/self-ssl.key and ./nginx/ssl/self.cert.
  + 