* Access the Guacamole interface with “[IP of VM]:[port assigned to the ‘guacamole’ container]/guacamole”. So for me it would be **192.168.0.138:8080/guacamole**
* Log in with “guacadmin:guacadmin”
* Now to create a connection to another container, fill out the following fields:
  + Name: choose a name
  + Protocol: SSH
  + Parameters -> Network -> Hostname: name of the container that you want to SSH into
  + Parameters -> Network -> Port: 22 (SSH default port)
  + Parameters -> Authentication -> Username: root
  + Parameters -> Authentication -> Password: root

***Useful commands:***

* + **sudo nano docker-compose.yaml**
    - Edit the docker compose file, or create it if it doesn’t exist
  + **sudo docker compose up -d**
    - Start all of the containers defined in the yaml file
  + **sudo docker compose stop**
    - Stops all running containers defined in the yaml file
  + **sudo docker compose exec --user root [name of container] bash**
    - Provides a shell to whichever container you specify, as root. Useful for debugging.
    - **sudo docker exec -u 0 -it [name of container] bash**
      * For non-docker compose containers (standalone docker containers created manually, without a yaml file).
  + **sudo docker compose logs [name of container]**
    - Displays logs of specified container. Useful for debugging.
  + **sudo docker ps -a**
    - Check all containers and various information about each one (debugging)
    - **sudo docker rm [container name/id]**
      * Remove an old, outdated container
  + **sudo docker image ls**
    - View all currently saved Docker images. Sometimes when trying to rebuild an image, it tries to write to some of the same locations that the old one uses, causing an error. Use this to see and delete (below) conflicting images.
    - **sudo docker image rm [image name/id]**
      * Remove an image
  + **sudo docker container inspect [name of container] --format "{{json .NetworkSettings.Ports}}"**
    - Check specified container’s open ports