## Connect to the server

1. Connect to the server with the SHH key. (Open ssh folder on your OS via terminal and type

ssh -i .\leadit\_prod\_private\_OpenSSH.ppk [root@178.62.235.35](mailto:root@178.62.235.35))

1. sudo su affiliedge – to switch from root user to affiliedge user
2. cd ../home/affiliedge/thunder – to go to the project directory

## Docker commands

* docker ps – display all running containers
* docker top <ID> - show running processes for selected container
* sudo kill -9 <PID> - killing process in container
* docker-compose exec app bash – enter the app container
* docker-compose exec mysql bash – enter the MySQL container
* mysql –u root –p - to enter the MySQL shell (first you need to enter DB pass and you need to be inside MySQL container)

## Server commands

Run them outside of the container, from inside the thunder directory.

* docker-compose exec -d app php artisan rabbitmq:consume – starting the queue
* docker-compose exec app php artisan route:cache
* docker-compose exec app php artisan config:clear
* docker-compose exec app php artisan cache:clear
* docker-compose exec app php artisan test – running tests (DO NOT DO THIS IN PRODUCTION, there are 100+ tests for now, I will add more tests if time allows)

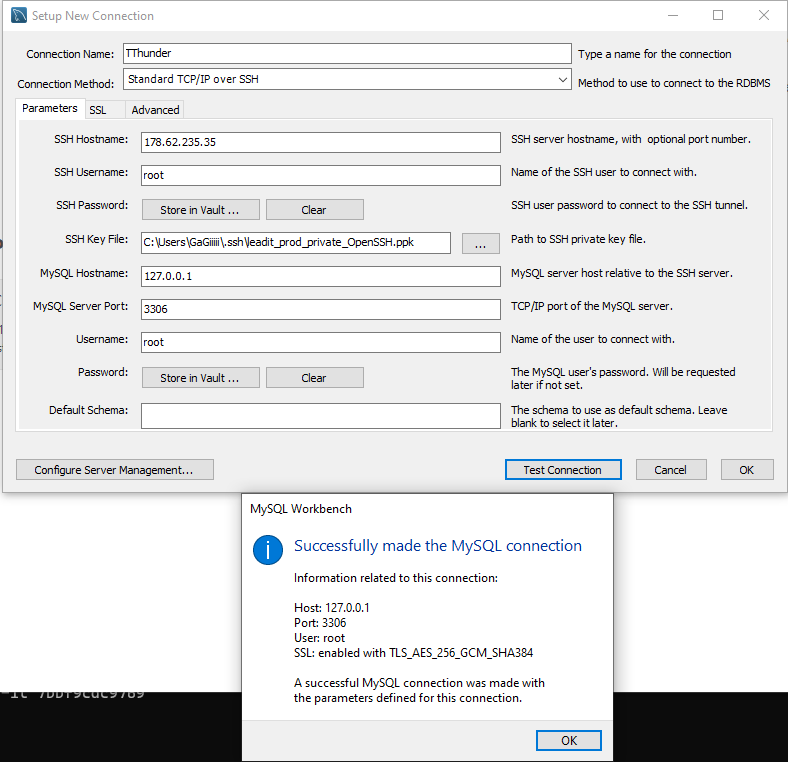
To start the cron job, enter the app container bash and type “cron” in the bash.

## Connect to the DB with MySQL Workbench

Select connection method “Standard TCP/IP over SSH”

Fill the form as shown:

* SSH pass is “thor”
* DB pass is “ezD53Uvc2dqnQ9kgLmGu” for root user or “rurph6SXBwaEUbTHxxWC” for affiliedge user



## Architecture

LeadIt app is a project built on top of Laravel and Vue.js frameworks. Laravel provides a suitable backend environment, while Vue is responsible for the frontend part. PHP version for this project is 7.4.

LeadIt app is being hosted on Digital Ocean and running on Docker. There are 4 main containers: app (where our PHP application lives), MySQL, Nginx, and Rabbitmq. To configure settings for these containers we use Dockerfile and docker-compose.yml.

APP CONTAINER

- here lives our application

- PHP 7.4

- composer

- artisan commands

- cron

- rabbitmq worker

NGINX CONTAINER

- simple Nginx conf

- currently only HTTP, we need certificates for HTTPS

- port 80

MYSQL CONTAINER

- simple MySQL config

- our database lives here

RABBITMQ CONTAINER

- message broker

- it is responsible for asynchronous messaging

- settings are created through the definitions.json file (users, permissions, queues) on the build

- currently, only default queue (in the future we will create dedicated queues)