

# Installing local "ScheduleDB" docker containers

- [Build and Run ScheduleDB Locally](#)
- [Configure Access to Scheduledb Services](#)
- [Accessing Scheduledb Services](#)

## Build and Run ScheduleDB Locally

How to load "ScheduleDB" using Docker containers locally on your system.

First you need to obtain access to the Git repository on Technology Nursery for use by TSSG members, send [ralph@navarrocomputing.com](mailto:ralph@navarrocomputing.com) your SSH public key by attaching it to an email and he will let you know when your ssh key has been posted to the git server.

Instructions on locating or creating your SSH public key can be found in:  
[Connecting to GitHub with SSH](#)

For more detail refer to the [TSSG Git Repository Overview](#).

Steps to recreate TssgTech:

1. Change directory to your local project repository.  
(my repo is ~/Projects verses ~/ideaProjects)  
**cd ~/Projects**
2. If you want to start fresh,  
cleanup any old local repository by deleting your ~/Projects/tssgTech  
using either the Docker Toolbox Interactive Shell (Quickstart Terminal) on Windows.  
**rm -rf tssgTech**
3. Clone the remote repository to a local copy  
**git clone ssh://git@technologynursery.org:2222/git-server/repos/tssgTech.git**
4. Change to your local repository directory.  
**cd tssgTech**
5. See the status of your local repository.  
The results should show that you are up-to-date with the *origin/master* branch.  
**git status**
6. Get a list of the available branches that are on the remote git server.  
**git branch --all**
7. Branch "remotes/origin/scheduledb" is what we want to checkout.  
Git will automatically checkout remotes/origin/scheduledb when you checkout scheduledb.  
**git checkout scheduledb**
8. You don't need to pull the latest changes because you just did a clone and checkout of a branch.  
However, a pull (after verifying you're on the right branch) should be done at least once a day – or more if others are committing frequently.  
**git pull**
9. Now that you have a local copy of the repository, and have checked out the scheduledb branch,

you can build a local instance of the development environment with all of the micro-services,  
by just running this script. Currently the script must be invoked from within the subdirectory it resides in.

```
cd ~/Projects/tssgTech/tools/docker/scheduledb  
./scheduledb.install
```

## Configure Access to Scheduledb Services

1. Determine your host IP address.
  - For Docker Toolbox on Windows
    - Use the IP address from 'docker-machine ip'
  - For Docker Desktop on Windows
    - Use the PC's IP Address
  - For Linux Local
    - Use the PC's IP Address
  - For Mac
    - # [Pushparathi Pandian](#), please specify. [Ralph A. Navarro Jr.](#) Added the below instructions
    - Click on the WiFi menu button in your **Mac's** menu bar.
    - Then select Network Preferences. ...
    - Next, select your network. ...
    - Then click on the Advanced button. ...
    - Next, click the TCP/IP tab. ...
    - Your computer's **IP address** will be a series of numbers next to **IPv4 Address**.
2. Locate your hosts file.
  - For Windows
    - c:\Windows\System32\etc\hosts
  - For Linux
    - /etc/hosts
  - For Mac
    - /etc/hosts    # [Pushparathi Pandian](#), please confirm. [Ralph A. Navarro Jr.](#) confirmed
3. As admin or root, add a line to your hosts file.

- <host IP>      website.sdb backend.sdb frontend.sdb mongo-express.sdb

## Accessing Scheduledb Services

You should now be able to access scheduledb services with your browser:

- <http://backend.sdb/meetings/webSchedule/WedGenMtg>
- <http://website.sdb/schedule.html>
- <http://frontend.sdb/user/login>
- <http://mongo-express.sdb/db/tssg-tech/>