Install Instructions

To get the server running you need to have a machine with a publicly-accessible IP address. The target server should be running ubuntu linux, ideally version 16.04. Once the machine is running, you should configure public-key based authentication and disable password-based authentication. Once the machine is figured to meet your specifications, run the provision script with the following command `sh server/provision/provision.sh`. This script installs a few necessary pieces of software, most notably NodeJS (the server base) and MongoDB (the database). Once that is complete, you can start up the server by running `npm start` from within the server/ directory. If you wish to use a process manager service, such as pm2 which is installed in the provision script, run `pm2 start server.js` instead (http://pm2.keymetrics.io/).

NodeJS

To read about NodeJS or to download it manually, visit https://nodejs.org/en/. It is a lightweight javascript-based server that allows the developer to iterate and improve quickly. By default, the provision script will install the current version of NodeJS.

Installing Server Dependencies

To install the third-party modules required to run the server, run the command `npm install` from the /server/ directory. This will automatically install of the dependencies listed in the package.json file.

Server Maintenance Instructions

Updating modules

To update third-party modules, an admin can simply run the `npm install` command from the /server/ directory. This will update all of the package.json dependencies to their most recent version. To update to the most recent version of NodeJS, an admin can simply follow the provision steps again and rerun the provision script. For official NodeJS instructions, see here: https://nodejs.org/en/download/package-manager/#debian-and-ubuntu-based-linux-distributions

Accessing Database

To access the database directly, an admin with access to the web/database server can open a shell to interface with MongoDB by entering `mongo` into the terminal. From there, the admin can enter a variety of MongoDB commands to insert, update, or delete records in the database (https://docs.mongodb.com/manual/reference/mongo-shell/). The default database name that the server uses is 'storkd'.

Logging

To view the access logs for the server, view the file 'access.log' under the server user's home directory. The logs are stored in standard Apache format, and can be cleared by simply deleting the

file. To perform more detailed analysis on the logs, the file would likely need to be put into a database or tokenized before being used with an analytics service.

Uploaded Files

By default, all uploaded files are stored to the /public/ directory, where they are then served statically. This approach is functional for small to medium amounts of traffic and upload capacity, but is not sufficient to sustain a 1000+ user base. If that point is reached, a more dedicated content delivery system will have to be implemented, such as Amazon Web Services S3. This will offload all of the uploaded images to that location where they can be served. To implement this change, a developer will have to modify the storage location of the 'multer' third-party NodeJS module that is currently being used.

Configuration

If the user base becomes large enough that the app requires a dedicated database server (separate from the web and app server), the only change that needs to be made is in the /server/app/config/config.js file. The 'database' item contains the url location of the database. Since the database server currently resides on the same machine as the app server, it is only accessible locally and thus does not need a password. If the database server is hosted separately, it will need to be password protected to prevent intruders from being able to connect to it directly.