

Deployment Documentation

Client

1. Update "SERVER_ROOT" constant in the CONSTANTS file to reflect the address of the server with valid SSL certificates
2. Deploy and run the mobile client locally - ([Client](#)) [How to install and run the mobile client](#)
3. Publish the application to the internet - ([Client](#)) [How to deploy and publish the application](#)

Server Requirements

1. Ubuntu 16.04 LTS
2. Public IP with valid DNS name with following publicly open network ports:
 - a. 443 : HTTPS service endpoints
 - b. 22 : SSH access
 - c. 8080 : Adminer (for development purposes)
 - d. 8099 : Server logs
3. Valid SSL certificates
4. Docker Engine <https://docs.docker.com/engine/>
5. Docker-compose <https://docs.docker.com/compose/>
6. Google account with full access (for email sending capability)

Installation Steps

1. Git clone
2. Install SSL certificates <https://certbot.eff.org/lets-encrypt/ubuntuxenial-other>
 - Once valid SSL certificates are obtained and copied to the server machine, update the location to these SSL certificates ('privkey.pem' and 'fullchain.pem') in

```
server/Scripts/installSSLCerts.sh : $SSL_CERT_DIR
```

3. Install docker and docker-compose.
 - Scripts to install 'Docker Engine' and 'docker-compose' are provided:

```
sudo ./installDocker-compose.sh
sudo ./installDocker.sh
```

Alternatively, they can be downloaded from the links provided under requirements.

4. Environment variables
 - a. Setting up Google email credentials. A Google account was created to have this functionality. As such, any other google account will also work. (Steps at the end)
 - Update the **server/apiContainer/register/.env** file inside with the following contents:

```
SERVICE=gmail
EMAIL=plain-text-email-id-here
PASSWORD=plain-text-password-here
```

- b. Setting up PostgreSQL environment variables

- Update **server/serviceContainer/common/.env.docker** with the following contents:

```
DB_USER = "your_database_user"
DB_NAME = " your_database_name"
DB_PASSWORD = "your_database_password"
```

- Update **server/serviceContainer/database/docker-compose.yml** with the following contents:

```
environment:
  POSTGRES_USER: your_database_user
  POSTGRES_PASSWORD: your_database_password
  POSTGRES_DB: your_database_name
```

Running the server

Once all requirements have been installed, run the following script from server/

```
sudo ./runServer.sh
```

To list running services

```
sudo ./Scripts/inspectImages.sh
```

Steps performed by runSever.sh:

1. Install the SSL certificates into the API container services.
2. Stop any existing instances of the server
3. Mount local volume for PostgreSQL database
4. Start 'Service container' along with Database container
5. Start 'API container'
6. Start logging service

3 API endpoints are available for mobile clients:

- POST: <https://server-dns/register/user>
- POST: <https://server-dns/write/user>
- GET : <https://server-dns/sync/user>

Allowing google access to email client

A Google account was created to implement the Email functionality. For this account, security settings have already been configured. Alternatively, any other google account or a new one can also be used, provided the settings are configured correctly. https://myaccount.google.com/lesssecureapps?pli=1&rapt=AEjHL4PHMzs-n5ZcK0DNjy9GQr2mAijUw7o_hlJBUGc6WDthA4zmSz8XW4AYJ9szOQOVnc2j3S2fDFVG0KCO7ZaKDDH8W1lg **Ensure that your browser is logged in with the correct credentials**

Due to google security policies, account access by an application is blocked by default on all new machines. Default access cannot be provided by Google, it can only be requested by the Application. Hence, this can only be generated once the server is running and it attempts to send an email.

In order to validate the server-machine as a valid user, we need to generate the unlockCaptcha. **This needs to be done only once, for every new machine that the server is ported to.**

After starting the server, the first request to validate an MP will generate the login attempt. Once this error is generated with a link (<https://accounts.google.com/b/0/DisplayUnlockCaptcha>) click on the link. **Ensure that your browser is logged in with the correct credentials.** After successfully verifying the new server machine, email capability will be accessible