

Gogs

With Gogs' ability to authenticate over SMTP it is trivial to integrate it with openemail. Few changes are needed:

1. Open `docker-compose.override.yml` and add Gogs:

```
version: '2.1'
services:

  gogs-openemail:
    image: gogs/gogs
    volumes:
      - ./data/gogs:/data
    networks:
      openemail-network:
        aliases:
          - gogs
    ports:
      - "${GOGS_SSH_PORT:-127.0.0.1:4000}:22"
```

2. Create `data/conf/nginx/site.gogs.custom`, add:

```
location /gogs/ {
    proxy_pass http://gogs:3000/;
}
```

3. Open `openemail.conf` and define the binding you want Gogs to use for SSH. Example:

```
GOGS_SSH_PORT=127.0.0.1:4000
```

5. Run `docker-compose up -d` to bring up the Gogs container and run `docker-compose restart nginx-openemail` afterwards.

6. Open `http://${OPENEMAIL_HOSTNAME}/gogs/`, for example `http://mx.example.org/gogs/`. For database details set `mysql` as database host. Use the value of `DBNAME` found in `openemail.conf` as database name, `DBUSER` as database user and `DBPASS` as database password.

7. Once the installation is complete, login as admin and set "settings" -> "authorization" -> "enable SMTP". SMTP Host should be `postfix` with port `587`, set `Skip TLS Verify` as we are using an unlisted SAN ("postfix" is most likely not part of your certificate).

8. Create `data/gogs/gogs/conf/app.ini` and set following values. You can consult [Gogs cheat sheet](#) for their meaning and other possible values.

```
[server]
SSH_LISTEN_PORT = 22
# For GOGS_SSH_PORT=127.0.0.1:4000 in openemail.conf, set:
SSH_DOMAIN = 127.0.0.1
SSH_PORT = 4000
# For OPENEMAIL_HOSTNAME=mx.example.org in openemail.conf (and default
ports for HTTPS), set:
ROOT_URL = https://mx.example.org/gogs/
```

9. Restart Gogs with `docker-compose restart gogs-openemail`. Your users should be able to login with openemail managed accounts.