

```
docker pull gitea/gitea:latest
docker pull gitea/gitea:linux-arm64-rootless
docker pull gitea/gitea:linux-arm64
docker pull gitea/gitea:linux-amd64
docker pull gitea/gitea:1.14.0
docker pull gitea/gitea:1.14
docker pull gitea/gitea:1
```

```
docker pull gitea/gitea
```

```
version: '2'
services:
```

```
  web:
```

```
    image: gitea/gitea:1.12.4
```

```
    volumes:
```

```
      - ./data:/data
```

```
    ports:
```

```
      - "3000:3000"
```

```
      - "22:22"
```

```
    depends_on:
```

```
      - db
```

```
    restart: always
```

```
  db:
```

```
    image: mariadb:10
```

```
    restart: always
```

```
    environment:
```

```
      - MYSQL_ROOT_PASSWORD=changeme
```

```
      - MYSQL_DATABASE=gitea
```

```
      - MYSQL_USER=gitea
```

```
      - MYSQL_PASSWORD=changeme
```

```
    volumes:
```

```
      - ./db:/var/lib/mysql
```

## Basics

```
version: "3"

networks:
  gitea:
    external: false

services:
  server:
    image: gitea/gitea:1.14.0
    container_name: gitea
    environment:
      - USER_UID=1000
      - USER_GID=1000
    restart: always
    networks:
      - gitea
    volumes:
      - ./gitea:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "222:22"
```

## Ports

```
version: "3"

networks:
  gitea:
    external: false

services:
  server:
    image: gitea/gitea:1.14.0
    container_name: gitea
    environment:
      - USER_UID=1000
      - USER_GID=1000
    restart: always
    networks:
      - gitea
    volumes:
      - ./gitea:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
```

```
- - "3000:3000"
- - "222:22"
+ - "8080:3000"
+ - "2221:22"
```

## Databases

### MySQL database

version: "3"

```
networks:
  gitea:
    external: false
```

```
services:
  server:
    image: gitea/gitea:1.14.0
    container_name: gitea
    environment:
      - USER_UID=1000
      - USER_GID=1000
+   - GITEA__database__TYPE=mysql
+   - GITEA__database__HOST=db:3306
+   - GITEA__database__NAME=gitea
+   - GITEA__database__USER=gitea
+   - GITEA__database__PASSWD=gitea
    restart: always
    networks:
      - gitea
    volumes:
      - ./gitea:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "222:22"
+   depends_on:
+     - db
+   db:
+     image: mysql:5.7
+     restart: always
+     environment:
+       - MYSQL_ROOT_PASSWORD=gitea
+       - MYSQL_USER=gitea
+       - MYSQL_PASSWORD=gitea
+       - MYSQL_DATABASE=gitea
+     networks:
+       - gitea
+     volumes:
+       - ./mysql:/var/lib/mysql
```

## PostgreSQL database

version: "3"

networks:

gitea:  
external: false

services:

server:

image: gitea/gitea:1.14.0

container\_name: gitea

environment:

- USER\_UID=1000

- USER\_GID=1000

+ - GITEA\_\_database\_\_TYPE=postgres

+ - GITEA\_\_database\_\_HOST=db:5432

+ - GITEA\_\_database\_\_NAME=gitea

+ - GITEA\_\_database\_\_USER=gitea

+ - GITEA\_\_database\_\_PASSWD=gitea

restart: always

networks:

- gitea

volumes:

- ./gitea:/data

- /etc/timezone:/etc/timezone:ro

- /etc/localtime:/etc/localtime:ro

ports:

- "3000:3000"

- "222:22"

+ depends\_on:

+ - db

+

+ db:

+ image: postgres:9.6

+ restart: always

+ environment:

+ - POSTGRES\_USER=gitea

+ - POSTGRES\_PASSWORD=gitea

+ - POSTGRES\_DB=gitea

+ networks:

+ - gitea

+ volumes:

+ - ./postgres:/var/lib/postgresql/data

## Named volumes

version: "3"

```
networks:
  gitea:
    external: false
```

```
+volumes:
+  gitea:
+    driver: local
+
services:
  server:
    image: gitea/gitea:1.14.0
    container_name: gitea
    restart: always
    networks:
      - gitea
    volumes:
-     - ./gitea:/data
+     - gitea:/data
      - /etc/timezone:/etc/timezone:ro
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "3000:3000"
      - "222:22"
```

Configure the user inside Gitea using environment variables

- USER: **git**: The username of the user that runs Gitea within the container.
- USER\_UID: **1000**: The UID (Unix user ID) of the user that runs Gitea within the container. Match this to the UID of the owner of the /data volume if using host volumes (this is not necessary with named volumes).
- USER\_GID: **1000**: The GID (Unix group ID) of the user that runs Gitea within the container. Match this to the GID of the owner of the /data volume if using host volumes (this is not necessary with named volumes).

## Customization

Customization files described [here](#) should be placed in /data/gitea directory. If using host volumes, it's quite easy to access these files; for named volumes, this is done through another container or by direct access at /var/lib/docker/volumes/gitea\_gitea/\_data. The configuration file will be saved at /data/gitea/conf/app.ini after the installation.

# Upgrading

**Make sure you have volumed data to somewhere outside Docker container** ☐☐

To upgrade your installation to the latest release:

```
# Edit `docker-compose.yml` to update the version, if you have one specified
# Pull new images
docker-compose pull
# Start a new container, automatically removes old one
docker-compose up -d
```

## Managing Deployments With Environment Variables

In addition to the environment variables above, any settings in app.ini can be set or overridden with an environment variable of the form:

GITEA\_\_SECTION\_NAME\_\_KEY\_NAME. These settings are applied each time the docker container starts. Full information [here](#).

These environment variables can be passed to the docker container in docker-compose.yml. The following example will enable an smtp mail server if the required env variables GITEA\_\_mailer\_\_FROM, GITEA\_\_mailer\_\_HOST, GITEA\_\_mailer\_\_PASSWD are set on the host or in a .env file in the same directory as docker-compose.yml:

```
...
services:
  server:
    environment:
      - GITEA__mailer__ENABLED=true
      - GITEA__mailer__FROM=${GITEA__mailer__FROM:?GITEA__mailer__FROM not
set}
      - GITEA__mailer__MAILER_TYPE=smtp
      - GITEA__mailer__HOST=${GITEA__mailer__HOST:?GITEA__mailer__HOST not set}
      - GITEA__mailer__IS_TLS_ENABLED=true
      - GITEA__mailer__USER=${GITEA__mailer__USER:-apikey}
      - GITEA__mailer__PASSWD="""${GITEA__mailer__PASSWD:?
GITEA__mailer__PASSWD not set}"""
```

To set required TOKEN and SECRET values, consider using gitea's built-in [generate utility functions](#).

## SSH Container Passthrough

Since SSH is running inside the container, SSH needs to be passed through from the host to the container if SSH support is desired. One option would be to run the container SSH on a non-standard port (or moving the host port to a non-standard port). Another option which might be more straightforward is to forward SSH connections from the host to the container. This setup is explained in the following. This guide assumes that you have created a user on the host called git which shares the same UID/ GID as the container values USER\_UID/ USER\_GID. These values can be set as environment variables in the docker-compose.yml:

environment:

- USER\_UID=1000
- USER\_GID=1000

Next mount /home/git/.ssh of the host into the container. Otherwise the SSH authentication cannot work inside the container.

volumes:

- /home/git/.ssh:/data/git/.ssh

Now a SSH key pair needs to be created on the host. This key pair will be used to authenticate the git user on the host to the container.

```
sudo -u git ssh-keygen -t rsa -b 4096 -C "Gitea Host Key"
```

In the next step a file named /app/gitea/gitea (with executable permissions) needs to be created on the host. This file will issue the SSH forwarding from the host to the container. Add the following contents to /app/gitea/gitea:

```
ssh -p 2222 -o StrictHostKeyChecking=no git@127.0.0.1  
"SSH_ORIGINAL_COMMAND=\"${SSH_ORIGINAL_COMMAND}\" $0 $@"
```

To make the forwarding work, the SSH port of the container (22) needs to be mapped to the host port 2222 in docker-compose.yml . Since this port does not need to be exposed to the outside world, it can be mapped to the localhost of the host machine:

```
ports:  
  # [...]  
  - "127.0.0.1:2222:22"
```

In addition, /home/git/.ssh/authorized\_keys on the host needs to be modified. It needs to act in the same way as authorized\_keys within the Gitea container. Therefore add the public key of the key you created above ("Gitea Host Key") to ~/.ssh/authorized\_keys. This can be done via `echo "$(cat /home/git/.ssh/id_rsa.pub)" >> /home/git/.ssh/authorized_keys`. Important: The pubkey from the git user needs to be added "as is" while all other pubkeys added via the Gitea web interface will be prefixed with `command="/app [...]`.

The file should then look somewhat like

```
# SSH pubkey from git user  
ssh-rsa <Gitea Host Key>
```

```
# other keys from users
```

```
command="/app/gitea/gitea --config=/data/gitea/conf/app.ini serv key-1",no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty <user pubkey>
```

Here is a detailed explanation what is happening when a SSH request is made:

1. A SSH request is made against the host (usually port 22) using the git user, e.g. `git clone git@domain:user/repo.git`.
2. In /home/git/.ssh/authorized\_keys , the command executes the /app/gitea/gitea script.
3. /app/gitea/gitea forwards the SSH request to port 2222 which is mapped to the SSH port (22) of the container.
4. Due to the existence of the public key of the git user in /home/git/.ssh/authorized\_keys the authentication host → container succeeds and the SSH request get forwarded to Gitea running in the docker container.

If a new SSH key is added in the Gitea web interface, it will be appended to `.ssh/authorized_keys` in the same way as the already existing key.

### **Notes**

SSH container passthrough will work only if

- opensshd is used in the container
- if AuthorizedKeysCommand is not used in combination with `SSH_CREATE_AUTHORIZED_KEYS_FILE=false` to disable authorized files key generation
- `LOCAL_ROOT_URL` is not changed