

Moodle

- [How to install Moodle on Docker Compose](#)
- [Moodle SSL/DNS troubleshooting](#)

How to install Moodle on Docker Compose

If you don't have docker and [docker compose](#), do that now.

docker-compose.yml file looks like:

```
version: '2'

services:
  mariadb:
    image: docker.io/bitnami/mariadb:11.4
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
      - MARIADB_USER=bn_moodle
      - MARIADB_DATABASE=bitnami_moodle
      - MARIADB_CHARACTER_SET=utf8mb4
      - MARIADB_COLLATE=utf8mb4_unicode_ci
    volumes:
      - 'mariadb_data:/bitnami/mariadb'
  moodle:
    image: docker.io/bitnami/moodle:4.4
    ports:
      - '80:8080'
      - '443:8443'
    environment:
      - MOODLE_DATABASE_HOST=mariadb
      - MOODLE_DATABASE_PORT_NUMBER=3306
      - MOODLE_DATABASE_USER=bn_moodle
      - MOODLE_DATABASE_NAME=bitnami_moodle
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      - ALLOW_EMPTY_PASSWORD=yes
    volumes:
      - 'moodle_data:/bitnami/moodle'
```

```
- 'moodledata_data:/bitnami/moodledata'
depends_on:
  - mariadb
volumes:
  mariadb_data:
    driver: local
  moodle_data:
    driver: local
  moodledata_data:
    driver: local
```

Default password login info is... Change it ASAP.

Username: user

Pass: bitnami

Using a proxy server to use SSL? [go here...](#)

Moodle SSL/DNS troubleshooting

By default, Moodle needs to use a SSL cert on its own. If you choose to use a 3rd party proxy to do it for you, you will need to change some settings.

Find your config.php file. It will be wherever you put it. Mine is located documents/moodle/moodledata/config.php

```
<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype    = 'mariadb';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'mariadb';
$CFG->dbname    = 'bitnami_moodle';
$CFG->dbuser    = 'bn_moodle';
$CFG->dbpass    = '';
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
    'dbpersist' => 0,
    'dbport'    => 3306,
    'dbsocket'  => '',
    'dbcollation' => 'utf8mb4_unicode_ci',
);

if (empty($_SERVER['HTTP_HOST'])) {
    $_SERVER['HTTP_HOST'] = '127.0.0.1:8080';
}
if (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == 'on') {
    $CFG->wwwroot = 'https://' . $_SERVER['HTTP_HOST'];
} else {
    $CFG->wwwroot = 'http://' . $_SERVER['HTTP_HOST'];
}
```

```

}
$CFG->dataroot = '/bitnami/moodledata';
$CFG->admin     = 'admin';

$CFG->directorypermissions = 02775;

require_once(__DIR__ . '/lib/setup.php');

// There is no php closing tag in this file,
// it is intentional because it prevents trailing whitespace problems!

~

```

This will be good if you want to keep SSL cert and HTTPS managed by moodle. If not, and you would like it done by a proxy server like Apache or Nginx, this will not work. We will need to change a few things.

So, on the above, from line 20-34 we changed a few things so that it will look like the below.

I will be using Nginx so, i need to make Moodle look for that proxy by adding "\$CFG->sslproxy = 1;" to this config file. (line 21 in the below).

Then, also change the "\$CFG->wwwroot = 'https://yomammas.house.com';" to have your own domain name.

```

<?php // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype    = 'mariadb';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'mariadb';
$CFG->dbname    = 'bitnami_moodle';
$CFG->dbuser    = 'bn_moodle';
$CFG->dbpass    = ''; // Set your database password here
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
    'dbpersist' => 0,
    'dbport'    => 3306,
    'dbsocket'  => '',
    'dbcollation' => 'utf8mb4_unicode_ci',

```

```
);
```

```
$CFG->sslproxy = 1;
```

```
// Set the wwwroot to the HTTPS URL
```

```
$CFG->wwwroot = 'https://yomammas.house.com';
```

```
$CFG->dataroot = '/bitnami/moodledata';
```

```
$CFG->admin = 'admin';
```

```
$CFG->directorypermissions = 02775;
```

```
require_once(__DIR__ . '/lib/setup.php');
```

```
// There is no php closing tag in this file,
```

```
// it is intentional because it prevents trailing whitespace problems!
```

Now, you should be able to use your apache or nginx proxy to redirect and SSL for you, correctly. YAY! lets get learning.