

# Locally Hosting Moodle using Docker

Using the bitnami/moodle docker image. <https://hub.docker.com/r/bitnami/moodle>

Commands for Installation and setting up

```
$ docker pull bitnami/moodle
```

```
$ docker network create moodle-network
```

Create a volume for MariaDB persistence and create a MariaDB container

```
$ docker volume create --name mariadb_data
```

```
$ docker run -d --name mariadb \  
--env ALLOW_EMPTY_PASSWORD=yes \  
--env MARIADB_USER=bn_moodle \  
--env MARIADB_PASSWORD=bitnami \  
--env MARIADB_DATABASE=bitnami_moodle \  
--network moodle-network \  
--volume mariadb_data:/bitnami/mariadb \  
bitnami/mariadb:latest
```

Create a volume for Moodle persistence and launch container

```
$ docker volume create --name moodle_data
```

```
$ docker run -d --name moodle \  
-p 8080:8080 -p 8443:8443 \  
--env ALLOW_EMPTY_PASSWORD=yes \  
--env MOODLE_DATABASE_USER=bn_moodle \  
--env MOODLE_DATABASE_PASSWORD=bitnami \  
--env MOODLE_DATABASE_NAME=bitnami_moodle \  
--env MOODLE_USERNAME=admin \  
--env MOODLE_PASSWORD=admin123 \  
--env MOODLE_EMAIL=admin@example.com \  
--network moodle-network \  
--volume moodle_data:/bitnami/moodle \  
--volume moodledata_data:/bitnami/moodledata \  
bitnami/moodle:latest
```

Debug command to check if container is running correctly

```
$ docker ps
```

Accessing the application locally at <http://localhost:8080> or [http://<ip\\_addr>:8080/login/index.php](http://<ip_addr>:8080/login/index.php)

Command to shutdown the docker

```
$ docker stop moodle mariadb
$ docker rm moodle mariadb
$ docker volume rm moodle_data moodledata_data
```

Expected Output:

```
mariadb_data
moodle
mariadb
moodle
mariadb
moodle_data
moodledata_data
Mariadb_data
```

## Enable LTI Plugins in moodle

- Go to [Manage enrol plugins](#)  
Administration/Plugins/Enrolments/Manage enrol plugins and enable “Publish as LTI tool”
- Allow frame embedding in [HTTP security](#)

## Creating an LTI Plugin

Using the pyLTI. <https://pypi.org/project/PyLTI/> <https://github.com/mitodl/pylti>

Commands for Installation and setting up

```
$ pip install PyLTI flask setuptools
$ docker network create moodle-network
```

Run the flask server in github and open in browser <http://<your-IP>:5000/>

# Integrating LTI Plugin into moodle

Login to moodle as admin

In site administration , under plugins go to Activity modules / External tool / [Manage tools](#) / External tool configuration

## External tool configuration

☒ Tool settings

Tool name	<input type="text" value="my_lti_tool"/>
Tool URL	<input type="text" value="http://10.2.128.164:5000/launch"/>
Tool description	<input type="text" value="demo dummy prototype test poc"/>
LTI version	<input type="text" value="LTI 1.0/1.1"/>
Consumer key	<input type="text" value="moodle"/>
Shared secret	<input type="text" value="secret"/>
Custom parameters	<input type="text"/>

# Using the LTI Plugin in moodle

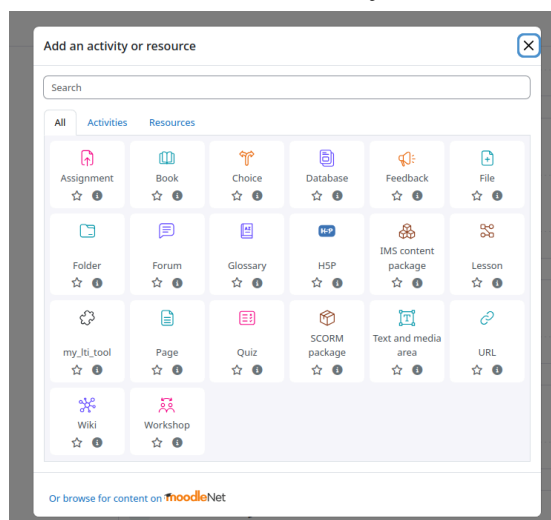
Login to moodle as admin

Go into a course or create a course.

Go to edit mode.

Add an activity.

In the selection menu, your custom plugin will show up.



Go back from edit mode and open the activity you just added (my-lti-tool).  
You will get a Welcome page as defined in templates/landing.html .