



"¿Qué me estás container?" Docker for dummies



Sara Arjona
@sara_arjona

Developer at Moodle HQ



Pau Ferrer
@crazyserver

Moodle Mobile developer
at Moodle HQ

#MootES18



ENTORNO DE DESARROLLO RÁPIDO CON VAGRANT

<http://goo.gl/4HPWpz>



ENTORNO DE DESARROLLO RÁPIDO CON VAGRANT

 moodlemoot
Mallorca 2015
#mootes15 #vagrant

Y EL AÑO QUE VIENE...

ENTORNO DE
DESARROLLO RÁPIDO
CON DOCKER



 moodle

Questions about Docker



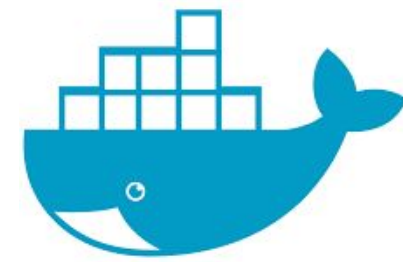
Who knows
about Docker?

Who uses
Docker for
development?

Who uses
Docker in
production?

Who tried but
could not do it?

Who is
interested in?

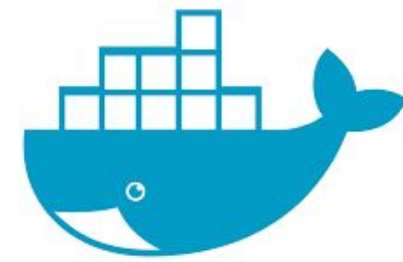


Docker

- Open platform for developing, shipping and running applications
- Allows package an application with all its dependencies into a standardized unit
- Provides abstraction of operating-system level virtualization
- Founded in 2013 as linux developer tool by Solomon Hykes (@solomonstre)

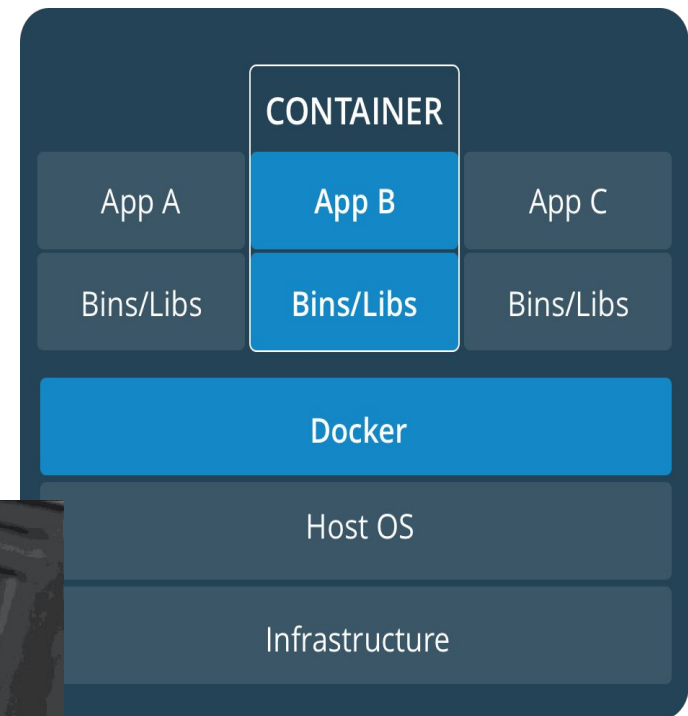
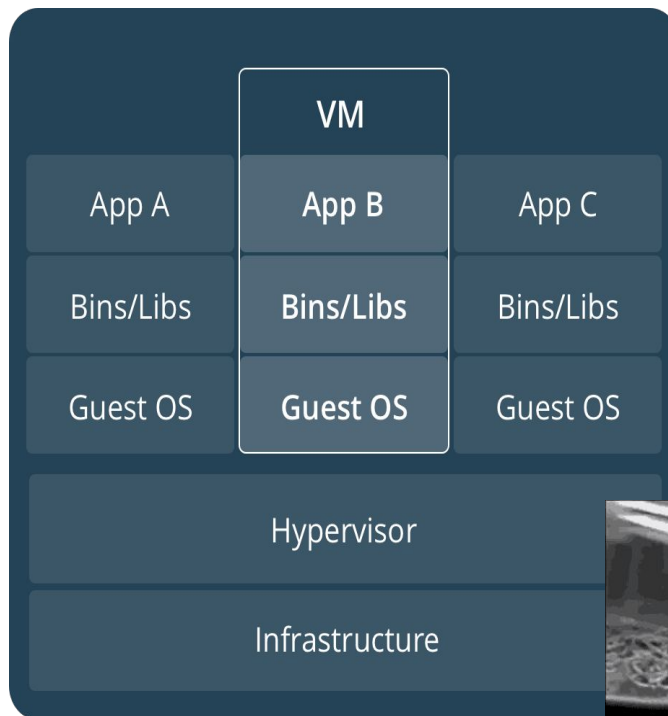


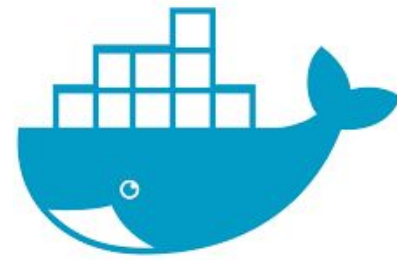
<https://www.docker.com/>



VM vs Container

- Containers are isolated but share OS and, when appropriate, bins/libraries
- There is no OS required inside of a container to run inside



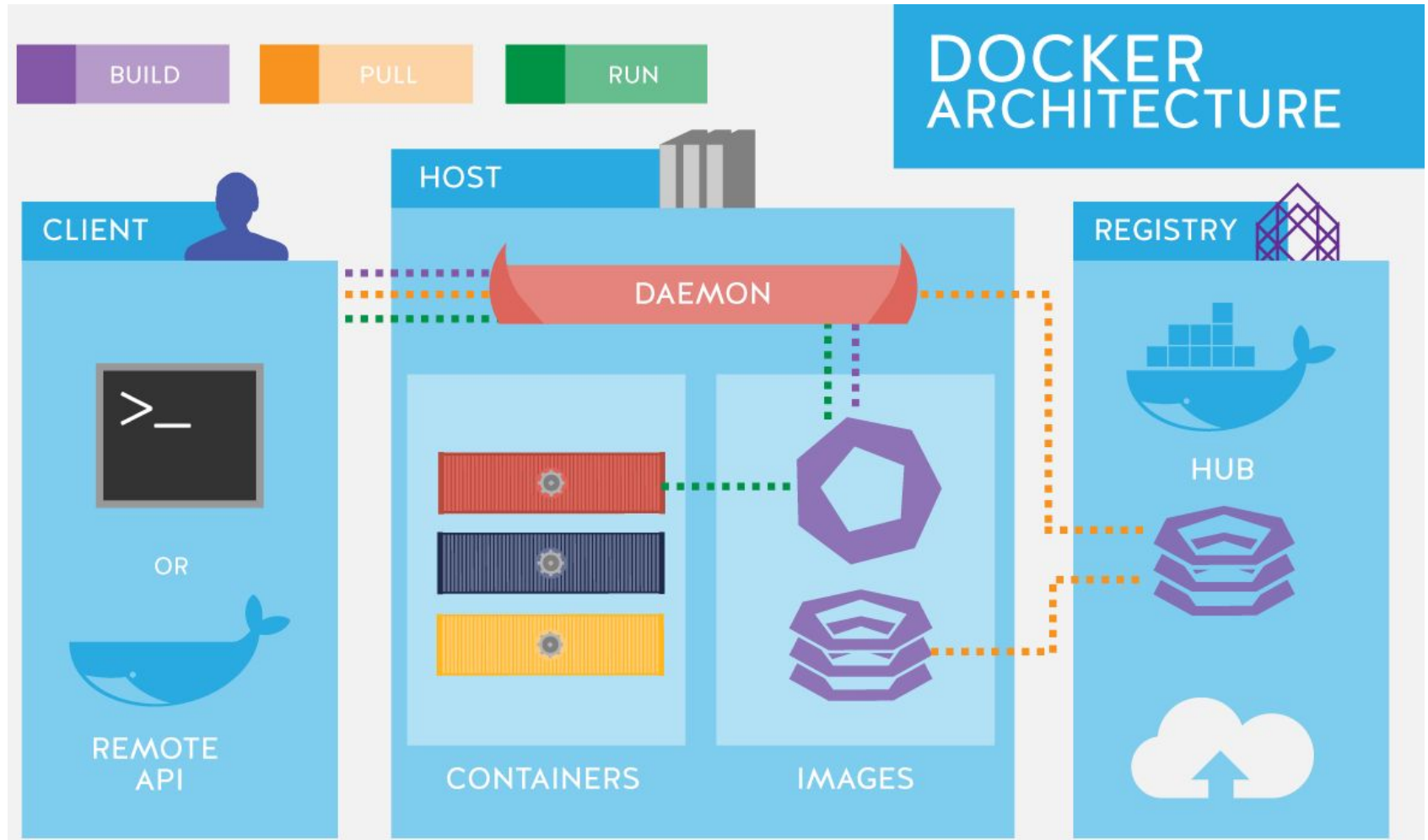
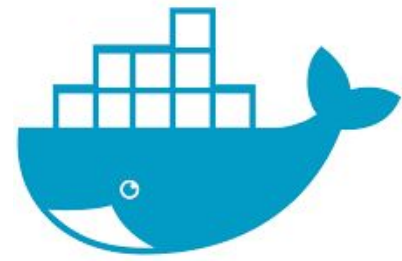


Docker benefits

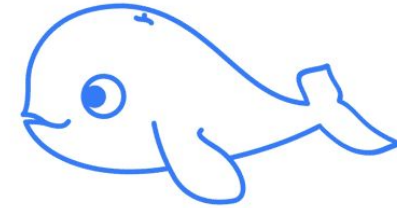
- Open Source
- Easy and fast deployment, migration, restarts
 - Microservices and integrations (APIs)
- Secure
- Lightweight (save disk, CPU & RAM)
 - No need to install guest OS
 - Run more apps on one host machine
- Great portability
 - Build in one environment, ship to another
- Simplify DevOps. Separation of concerns:
 - Developers focus on building apps
 - Sysadmins focus on deployment
- Version control capabilities



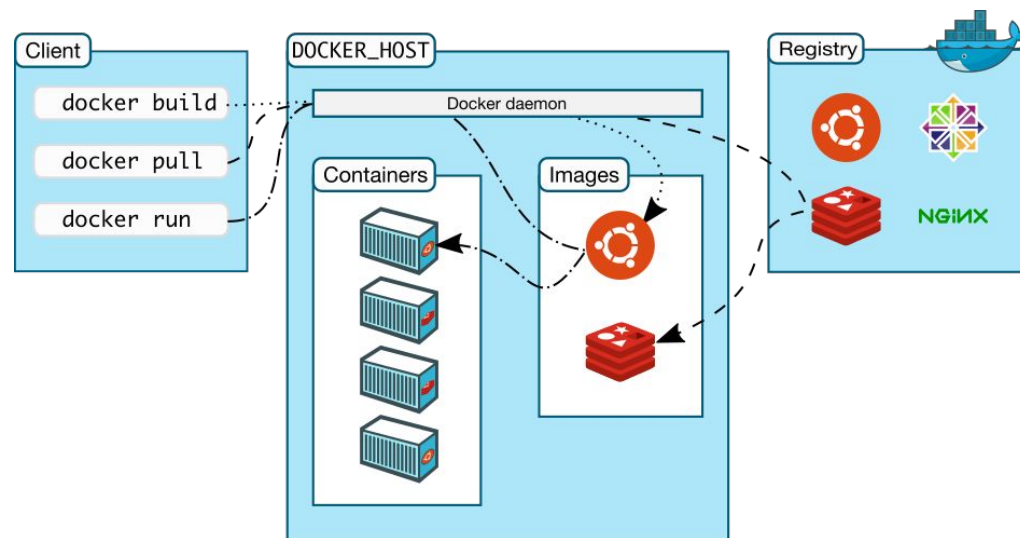
Components

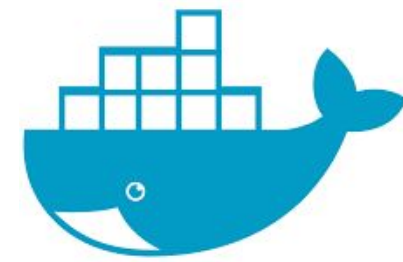


Docker workflow



- Pulls an image from the registry
- Creates a new container
- Allocates a filesystem and mounts a read-write layer
- Allocates a network/bridge interface
- Sets up an IP address
- Executes a process (if you specify it)
- Captures and provides application output





Who uses Docker?



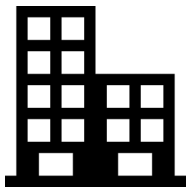
Developers

- Local development
- Development collaboration
- Sandbox environment (develop, test, debug, educate)
- To eliminate the “works on my machine” problem



System operators

- Run and manage apps faster
- Continuous Integration & Deployment
- Scaling apps
- Infrastructure configuration



Enterprises

- Continuous Delivery Model
- Ship new features faster and more securely





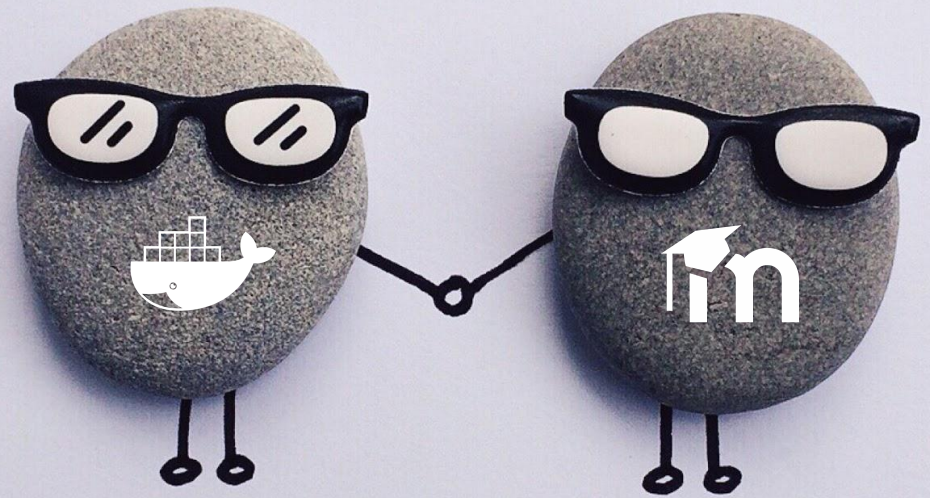
Interesting Docker tool

- **Portainer**
 - Simple management UI for Docker
 - <http://portainer.io>

The screenshot displays the Portainer web interface. On the left is a dark blue sidebar with navigation options: Dashboard, App Templates, Containers (selected), Images, Networks, Volumes, Events, Engine, and Portainer Settings (User management, Endpoints, Registries, Settings). The main content area is titled 'Container list' and shows a table of running containers. The table has columns for State, Name, Image, IP Address, Published Ports, and Ownership. All containers are in a 'running' state. Below the table are control buttons for Start, Stop, Kill, Restart, Pause, Resume, and Remove, along with an 'Add container' button and a 'Filter...' input field.

State	Name	Image	IP Address	Published Ports	Ownership
running	moodledocker_webserver_1	moodlehq/moodle-php-apache:7.1	172.18.0.11	8000:80	public
running	moodledocker_selenium_1	selenium/standalone-firefox-debug:2.53.1	172.18.0.10	5900:5900	public
running	moodledocker_ldap_1	larrycai/openldap	172.18.0.9	-	public
running	moodledocker_exttests_1	moodlehq/moodle-exttests	172.18.0.8	-	public
running	moodledocker_memcached1_1	memcached:1.4	172.18.0.7	-	public
running	moodledocker_solr_1	solr:6.5	172.18.0.6	-	public
running	moodledocker_memcached0_1	memcached:1.4	172.18.0.3	-	public
running	moodledocker_redis_1	redis:3	172.18.0.4	-	public
running	moodledocker_db_1	mysql:5	172.18.0.5	-	public
running	moodledocker_mailhog_1	mailhog/mailhog	172.18.0.2	-	public
running	portainer	portainer/portainer	172.17.0.2	9000:9000	public

Docker & Moodle





Moodle Docker features

- All supported database servers (PostgreSQL, MySQL, Microsoft SQL Server, Oracle XE)
- All supported PHP versions (5.6, 7.0, 7.1)
- Behat/Selenium configuration for Firefox and Chrome
- Catch-all smtp server and web interface to messages using [MailHog](#)
- All PHP Extensions enabled configured for external services (e.g. solr, ldap)
- Zero-configuration approach
- Backed by [automated tests](#)

<https://github.com/moodlehq/moodle-docker>



Moodle Docker configuration

```
# Clone moodle-docker repository
git clone https://github.com/moodlehq/moodle-docker.git
cd moodle-docker

# Set up path to Moodle code
export MOODLE_DOCKER_WWWROOT=/path/to/moodle/code

# Choose a db server: postgres, mariadb, mysql, mssql, oracle
export MOODLE_DOCKER_DB=mysql

# Ensure customized config.php for the Docker containers is in place
cp config.docker-template.php $MOODLE_DOCKER_WWWROOT/config.php
```

- Other environment variables:
 - MOODLE_DOCKER_PHP_VERSION (5.6, 7.0, 7.1)
 - MOODLE_DOCKER_BROWSER (for behat; default: firefox)
 - MOODLE_DOCKER_WEB_HOST (default: localhost)
 - MOODLE_DOCKER_WEB_PORT (default: 8000)
 - MOODLE_DOCKER_SELENIUM_VNC_PORT (e.g. 5900)
 - MOODLE_DOCKER_PHPUNIT_EXTERNAL_SERVICES (e.g. true)





Moodle Docker start up

```
# Start up containers
$ bin/moodle-docker-compose up -d
Creating network "moodledocker_default" with the default driver
Creating moodledocker_extttests_1 ...
Creating moodledocker_db_1 ...
Creating moodledocker_mailhog_1 ...
Creating moodledocker_selenium_1 ...
Creating moodledocker_mailhog_1
Creating moodledocker_extttests_1
Creating moodledocker_db_1
Creating moodledocker_selenium_1 ... done
Creating moodledocker_webserver_1 ...
Creating moodledocker_webserver_1 ... done

# Wait for DB to come up
$ bin/moodle-docker-wait-for-db
```



Moodle docker commands

- Stop containers

```
$ bin/moodle-docker-compose stop
```

- Start containers

```
$ bin/moodle-docker-compose start
```

- Shut down and destroy containers

```
$ bin/moodle-docker-compose down
```

- Access into a container

```
$ docker exec -it container_name bash
```



Moodle Docker for **manual** testing

- Initialize database

```
$ bin/moodle-docker-compose exec webserver php
admin/cli/install_database.php --agree-license
--fullname="Docker moodle" --shortname="docker_moodle"
--adminpass="test" --adminemail="admin@example.com"
```

- Default URL: <http://localhost:8000/>

System	Linux fced33d05230 4.13.0-45-generic #50~16.04.1-Ubuntu SMP Wed May 30 11:18:27 UTC 2018 x86_64
Build Date	Sep 8 2017 02:54:32
Configure Command	'./configure' '--build=x86_64-linux-gnu' '--with-config-file-path=/usr/local/etc/php' '--with-config-file-scan-dir=/usr/local/etc/php/conf.d' '--disable-cgi' '--enable-ftp' '--enable-mbstring' '--enable-mysqlnd' '--with-curl' '--with-libedit' '--with-openssl' '--with-zlib' '--with-pcre-regex=/usr' '--with-libdir=lib/x86_64-linux-gnu' '--with-apxs2' 'build_alias=x86_64-linux-gnu'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/usr/local/etc/php



Moodle Docker for **phpunit** testing

- Initialize phpunit environment

```
$ bin/moodle-docker-compose exec webserver php admin/tool/phpunit/cli/init.php
[...]  
++ Success ++  
  
PHPUnit test environment setup complete.
```

- Run tests

```
$ bin/moodle-docker-compose exec webserver vendor/bin/phpunit  
auth_manual_testcase auth/manual/tests/manual_test.php  
Moodle 3.6dev (Build: 20180614), 9e7c3978895c7cab585c2f5234ca536151d3bef6  
Php: 7.1.9, mysqli: 5.7.20, OS: Linux 4.13.0-45-generic x86_64  
PHPUnit 6.5.8 by Sebastian Bergmann and contributors.  
  
..                                                                    2 / 2 (100%)  
  
Time: 40.35 seconds, Memory: 36.00MB  
  
OK (2 tests, 7 assertions)
```



Moodle Docker for **behat** testing

- Initialize behat environment

```
$ bin/moodle-docker-compose exec webserver php admin/tool/behat/cli/init.php
[...]  
++ Success ++  
Acceptance tests site installed  
Acceptance tests environment enabled on http://webserver, to run the tests use:  
vendor/bin/behat --config /var/www/behatdata/behatrun/behat/behat.yml
```

- Run tests

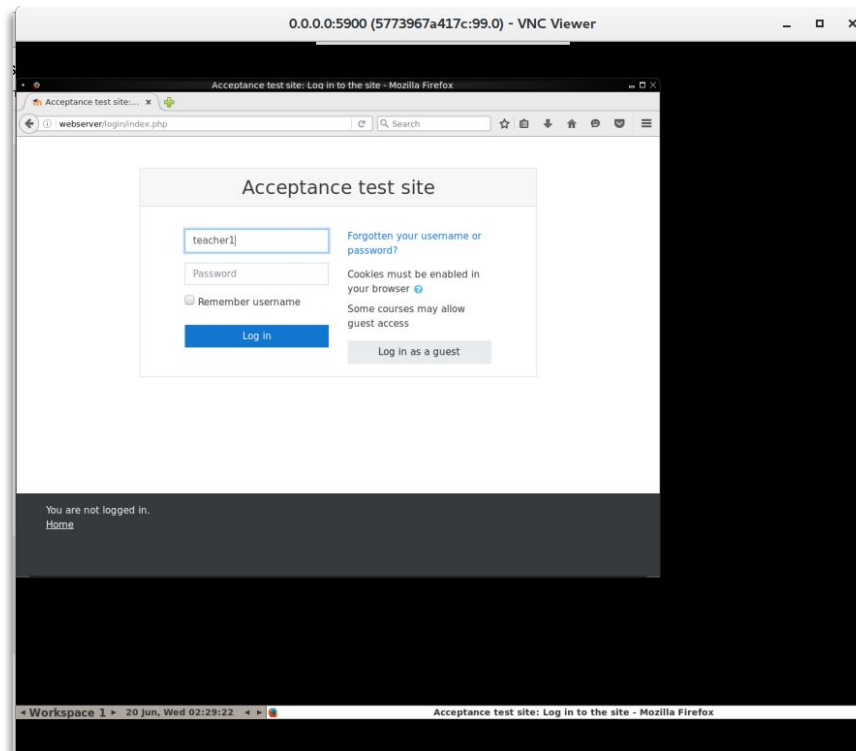
```
$ bin/moodle-docker-compose exec webserver php admin/tool/behat/cli/run.php  
--tags=@auth_manual  
Running single behat site:  
Moodle 3.6dev (Build: 20180614), 9e7c3978895c7cab585c2f5234ca536151d3bef6  
Php: 7.1.9, mysql: 5.7.20, OS: Linux 4.13.0-45-generic x86_64  
Server OS "Linux", Browser: "firefox"  
Started at 20-06-2018, 16:30  
.....  
  
2 scenarios (2 passed)  
15 steps (15 passed)  
0m55.51s (43.28Mb)
```

- The behat faildump directory is exposed at http://localhost:8000/_/faildumps/

Moodle Docker for **behat** testing



- Using VNC to view behat tests
 - a. Set MOODLE_DOCKER_SELENIUM_VNC_PORT (e.g. 5900)
 - b. Download a [VNC client](#)
 - c. Enter 0.0.0.0:5900 as the port in VNC Viewer (password 'secret')
 - d. Run behat tests





Moodle Docker for mailing

- Mailhog is listening on http://localhost:8000/_/mail
 - a. First time, don't forget to initialise 'No-reply address' setting

The screenshot shows the MailHog web interface in a browser window. The address bar shows 'localhost:8000/_/mail/#'. The interface includes a search bar, a refresh button, and a message list. The message list contains four entries:

From	Subject	Time	Size
"Admin User (via docker_moodle)" "User 1"	New message from Admin User	a few seconds ago	707 B
"Admin User (via docker_moodle)" "User 1"	Docker: account confirmation	a minute ago	1.96 kB
"Admin User (via docker_moodle)" "User 2"	Docker: account confirmation	3 minutes ago	1.96 kB
"Admin User (via stable_35)" "さくら 伊藤"	New message from Admin User	2 hours ago	787 B

On the left side, there is a 'Connected' status, an 'Inbox (4)' label, and a 'Delete all messages' button. Below that is a 'Jim' section with a description: 'Jim is a chaos monkey. Find out more at GitHub.' and an 'Enable Jim' button.



Moodle Docker for SOLR (search)

- MOODLE_DOCKER_PHPUNIT_EXTERNAL_SERVICES=true needed
- Create the 'moodle' index:

```
$ docker exec -it --user=solr moodledocker_solr_1 bin/solr
create_core -c moodle
[...]
Creating new core 'moodle' using command:
[...]
```

- Configure solr, set up search engine and index all the content:

Solr

Connection settings

Host name <small>search_solr server_hostname</small>	<input type="text" value="172.18.0.6"/>	Default: 127.0.0.1
Domain name of the Solr server.		
Index name <small>search_solr indexname</small>	<input type="text" value="moodle"/>	Default: Empty
Secure mode <small>search_solr secure</small>	<input type="checkbox"/> Default: No	
Port <small>search_solr server_port</small>	<input type="text" value="8983"/>	Default: 8983

Global search

▼ Search

Enter your search query ! ?

► Filter

There are required fields in this form marked !.

User 1

[View this result in context - in course Docker](#)



Moodle Docker for LDAP

- MOODLE_DOCKER_PHPUNIT_EXTERNAL_SERVICES=true needed
 - Used <https://github.com/larrycai/docker-openldap> image
- Check LDAP is working:

```
$ docker exec moodledocker_ldap_1 ldapsearch -x -H  
ldap://localhost -b ou=Users,dc=openstack,dc=org -LL
```

- Configure Authentication LDAP server:
 - Host URL: ldap://172.18.0.9* *Correct value could be checked, for instance, using portainer
 - Distinguished name: cn=admin,dc=openstack,dc=org
 - Password: password
 - Contexts: ou=users,dc=openstack,dc=org
 - User attribute: uid
 - Data mapping (First name): cn
 - Data mapping (Email address): mail
 - Data mapping (Description): description
 - Data mapping (Department): ou
- Users:
 - rjsmith / rJsmith
 - Larrycai / LarryCai

And the next
year...

:-P

The image shows a screenshot of the FORGE web interface. On the left is a dark sidebar with the FORGE logo and navigation options: 'Moodles', 'Create', 'Provisioning', and 'Containers'. The main content area is divided into two panels. The top panel, titled 'Current Moodle instances', features a sad face icon and the text 'We can't find any Moodle instances!' with a '+ CREATE A MOODLE INSTANCE' button. The bottom panel, titled 'Create a new Moodle instance', contains a form with the following fields and options:

- Subdomain*: `stable-master` (with a dropdown arrow) and `.moodledev.site`
- HTTP: `stable-master` (with a dropdown arrow) and `.moodledev.site`
- Absolute path to your Moodle source code*: `/home/sara/workspace/git/moodlehq/stable_master-dock`
- PHP: `PHP 7.1` (with a dropdown arrow)
- Database: `PostgreSQL` (with a dropdown arrow)
- Run containers after creation:
- Install Moodle:
- Create default users in Moodle:

At the bottom of the form are 'CANCEL' and 'CREATE' buttons. A progress indicator on the right shows the following steps:

- 1 Create docker containers
 - Creating a private network for the Moodle instance
 - Creating the database container
 - Creating the Moodle container
 - Connecting the database to the network
 - Connecting Moodle to the network
- 2 Start containers
- 3 Install Moodle



Sara Arjona
@sara_arjona

Developer at Moodle HQ



Pau Ferrer
@crazyserver

Moodle Mobile developer
at Moodle HQ

#MootES18