

Fase 4 - Activitat 8: Fase de Test del cicle de vida DevOps – Test de funcionament d'aplicacions PHP amb PHPUnit

0- Identificació del grup i activitat:

Curs: ASIX2

Projecte: PJ9 DevOps i Cloud Computing

Fase: 4

Activitat: 8

Grup/Individual: Individual

Membres/Alumne:

1- Introducció i objectius de l'activitat 8

- a) Comprendre el propòsit de l'activitat
- b) Lectura de les especificacions de l'activitat
- c) Desplegament d'aplicacions utilitzant contenidors Docker que tinguin incorporat el framework PHPUnit necessari per fer tests de funcionament.
- d) Execució dels tests de funcionament de l'aplicació i comprovació de resultats.
- e) Avaluar l'activitat.
- f) Execució dels tests de funcionament de l'aplicació i comprovació de resultats.

2- Creació de l'entorn de treball dins de la màquina física

- a) Accedeix al directori **pj9** → **f4** de la teva màquina **física** i crea:
 - Un directori de nom **a8**.
 - Dins del directori **a8** crea un directori de nom **vm** i un altre de nom **projectes**.
- b) Dins de **vm** desenvolupa un fitxer **Vagrantfile** per poder crear una màquina virtual amb les següents característiques:
 - Imatge de màquina virtual a descarregar: **debian/bookworm64**
 - Gestor de màquines virtuals: **virtualbox**
 - Número de CPUs de la màquina virtual: **3**
 - RAM de la màquina virtual = **3072MiB**
 - Nom de la màquina en VirtualBox: **pj9f4a8-xyyyz** (a xyyyz són les 2 primeres lletres dels teus noms i cognoms).
 - Nom de sistema de la màquina virtual: **pj9f4a8-xyyyz.fjeclot.local**
 - Carpeta de la màquina física compartida amb la virtual: **../projectes**
 - Carpeta de la màquina virtual compartida amb la física: **/home/vagrant/projectes**
 - Tipus de xarxa: **public_network** per treballar en mode **Bridge**.
 - Programari a instal·lar: **net-tools**, **whois**, **aptitude** i el programari per treballar amb **dockers**
- c) Posa en marxa la màquina virtual i comprova que funciona correctament i té les característiques demanades.
- d) Atura la màquina virtual utilitzant ordres de **vagrant**.
- e) Des de dins del directori **a8** crea un fitxer de nom **.gitignore** i fes que el seu contingut sigui:

```
.gitignore  
vm/.vagrant
```
- f) Amb **git**, fes un **add** i un **commit** dins del directori **a8**. El comentari del commit serà: **"Commit 1 pj9f4a8"**.

3- Creació de l'entorn de treball dins de la màquina virtual

- a) Posa en marxa i accedeix a la màquina virtual. Entra dins del directori **projectes**.
- b) Descarrega el projecte de nom **calculadora** del dipòsit: <https://github.com/asix2pj9/calculadora.git>. Fes els següents passos:
- Clona el dipòsit.
 - Entra dins del directori **calculadora** que s'ha creat i esborra el directori **.git**. Així, serà com si simplement haguessis descarregat els fitxers únicament.
- c) Comprova que dins del directori tens:
- Una carpeta de nom **app** amb els fitxers PHP i HTML de l'aplicació.
 - Una carpeta de nom **testPHPUnit** amb el script **testPHPUnit.sh** per fer test de funcionament.
 - Un fitxer **.dockerignore**
 - Un fitxer **Dockerfile**
 - Un fitxer **docker-compose.yml**
 - Un fitxer **nginx.conf**
- d) Posa en marxa el servei sCalculadora creant **5** contenidors amb l'aplicació i el distribuïdor de carrega. Executa: **docker compose up -d --scale sCalculadora=5**
- e) Comprova que:
- S'ha creat la imatge **icalculadora:1.0**. Executa: **docker images**
 - S'ha creat:
 - El servei **sCalculador**.
 - S'han creat **5** contenidors de l'aplicació.
 - S'ha creat el contenidor amb el distribuïdor de carrega **cProxyHTTP**.Executa: **docker compose ps -a**
- f) Observa quin port està utilitzant el servei **sCalculadora** i si aquest port s'ha publicat.
- g) Observa:
- El port publicat pel servei **sDistribueixCarrega**.
 - Cap a quin port rediregeix les connexions de la màquina virtual el servei **sDistribueixCarrega**
 - Quin servei hi ha darrera el port cap el qual rediregeix **sDistribueixCarrega** les connexions.
- h) Comprova que la màquina virtual mostra obert el port **8000/tcp**. Executa l'ordre: **sudo netstat -atupn**
- g) Comprova l'adreça IP de la màquina virtual. Executa: **ip -a show dev eth1**
- h) Accedeix des de la màquina física a l'aplicació a través de l'adreça IP de la màquina virtual i del port obert pel distribuïdor de carrega dins de la màquina virtual.

4- Modificacions de Dockerfile per incorporar PHPUnit a la imatge dels contenidors

4.1- Modificació de Dockerfile per incorporar PHPUnit a la imatge dels contenidors

Dins de la màquina virtual, modifiquem el fitxer **Dockerfile**. Anem a utilitzar una nova instrucció de creació d'imatge anomenada **RUN**. Aquesta instrucció executarà una ordre de sistema i incorporarà els seus resultats a la imatge. Actualitza **Dockerfile** afegint les següents instruccions al final:

```
# Incorporació del framework de tests de funcionament PHPUnit versió 12
RUN apt-get -y update
RUN apt-get -y install zip unzip curl
RUN curl -sS https://getcomposer.org/installer | php -- --install-dir=/usr/local/bin --
filename=composer
RUN composer require --dev phpunit/phpunit ^12
```

4.2- Modificació de l'aplicació per incorporar un test fet amb PHPUnit a la imatge dels contenidors

a) Comprova que dins del directori **app** de la màquina virtual hi ha un fitxer preparat per crear tests de funcionament de l'aplicació de nom **operacionsTest.php**. Aquest fitxer:

- S'incorporarà a tots els contenidors que hagi creat.
- Està preparat per fer els test de l'opció suma de l'aplicació.
- No està preparat per fer els tests de la resta d'opcions de l'aplicació.

b) Completa el fitxer **operacionsTest.php** per poder fer tests de resta, multiplicació, divisió i potenciació amb el següent joc de tests:

- Test per la Resta:
 - operand1 = 0, operand2 = 0
 - operand1 = 1, operand2 = 4
 - operand1 = 5, operand2 = 2
- Test per la Multiplicació:
 - operand1 = 0, operand2 = 0
 - operand1 = 2, operand2 = 4
 - operand1 = 5, operand2 = -2
 - operand1 = -2, operand2 = -3
- Test per la Divisió:
 - operand1 = 8, operand2 = 2
 - operand1 = 6, operand2 = -2
 - operand1 = -12, operand2 = -4
- Test per la Potenciació:
 - operand1 = 2, operand2 = 3
 - operand1 = 2, operand2 = -1
 - operand1 = -4, operand2 = -2

4.3- Modifica el fitxer docker-compose.yml per crear una nova versió de la imatge

Modifica també **docker-compose.yml** per crear una nova imatge de l'aplicació. Fes que la línia 3 tingui ara el següent contingut:

```
image: icalculadora:1.1
```

4.4- Posa novament en marxa l'aplicació

a) Posa novament en marxa el servei sCalculadora creant **5** contenidors amb l'aplicació i el distribuïdor de carrega. Executa: **docker compose up -d --scale sCalculadora=5**

b) Comprova que s'ha creat la nova imatge **icalculadora:1.1** executant: **docker images**

c) Observa quin port està utilitzant el servei **sCalculadora** i si aquest port s'ha publicat.

d) Observa:

- El port publicat pel servei **sDistribueixCarrega**.
- Cap a quin port rediregeix les connexions de la màquina virtual el servei **sDistribueixCarrega**
- Quin servei hi ha darrera el port cap el qual rediregeix **sDistribueixCarrega** les connexions.

e) Comprova que la màquina virtual mostra obert el port **8000/tcp**. Executa l'ordre: **sudo netstat -atupn**

f) Comprova l'adreça IP de la màquina virtual. Executa: **ip -a show dev eth1**

4.5- Fes el test de funcionament amb PHPUnit

- a) Fes que **testPHPUnit.sh** dins del directori **testPHPUnit** sigui executable.
- b) Executa **testPHPUnit.sh** → Dins del directori **testPHPUnit** escriu: **./testPHPUnit.sh**
- c) Comprova que el resultat és aquest per 5 contenidors:

```
vagrant@pj9f4a8-dacomo:~/projectes/calculadora/testPHPUnit$ ./testPHPUnit.sh
S'han trobat els següents contenidors:
calculadora-sCalculadora-5
calculadora-sCalculadora-4
calculadora-sCalculadora-3
calculadora-sCalculadora-2
calculadora-sCalculadora-1

=== Executant proves seqüencialment ===

⇒ Executant prova al contenidor: calculadora-sCalculadora-5 ...
calculadora-sCalculadora-5 - OK!!!

⇒ Executant prova al contenidor: calculadora-sCalculadora-4 ...
calculadora-sCalculadora-4 - OK!!!

⇒ Executant prova al contenidor: calculadora-sCalculadora-3 ...
calculadora-sCalculadora-3 - OK!!!

⇒ Executant prova al contenidor: calculadora-sCalculadora-2 ...
calculadora-sCalculadora-2 - OK!!!

⇒ Executant prova al contenidor: calculadora-sCalculadora-1 ...
calculadora-sCalculadora-1 - OK!!!

=== RESUM ===
Exitoses: 5
Fallides: 0
```

- d) Si hi ha cap error, comprova el codi de **operacionsTest.php** no té errors.
- e) Si l'error persisteix tot i ser correcte els valors que hem posat **operacionsTest.php** llavors, que estaria passant?

5- Modificacions de Dockerfile per incorporar PHPUnit a la imatge dels contenidors

- a) Des del directori a8 fes un add i un commit de tot el projecte. Fes que el comentari sigui: *"Commit 2 de pj9f4a8"*.
- b) Crea un dipòsit de **Github** de nom **pj9f4a8** i puja el teu projecte local al teu dipòsit de **Github**.

Lliurament de l'activitat

- a) Mostra'm l'aplicació funcionant
- b) Mostra'm el test de funciomanent sense errors.
- c) Mostra'm el teu dipòsit de Github
- d) Data límit per obtenir el 100% de la nota: **dijous 24-11-24** a les **19.10**. Posteriorment la nosta serà 0.