**ÍNDEX**

[Introducció 1](#_Toc197724201)

[Configuracions i desplegament amb Docker Compose (Fase 1) 1](#_Toc197724202)

[Introducció a Docker Compose 1](#_Toc197724203)

[Instal·lació i configuració de Docker Compose 2](#_Toc197724204)

[Creació d’un entorn multi-contenidor 4](#_Toc197724205)

[Orquestració i desplegament amb Docker Swarm (Fase 2) 13](#_Toc197724206)

[Introducció a Docker Swarm 13](#_Toc197724207)

[Configuració d’un clúster Swarm 13](#_Toc197724208)

[Migració del projecte de Docker Compose a Docker Swarm 14](#_Toc197724209)

[Seguretat a Docker Swarm (Fase 3) 17](#_Toc197724210)

[Tècniques de Seguretat 17](#_Toc197724211)

[Xifratge de Comunicacions amb TLS 17](#_Toc197724212)

[Orquestració amb Kubernetes (Fase 4) 19](#_Toc197724213)

[Introducció 19](#_Toc197724214)

[Creació i gestió de pods i serveis 19](#_Toc197724215)

[Deployment pel MySQL 20](#_Toc197724216)

[Deployment PhPMyAdmin 21](#_Toc197724217)

[Service per exposar el servidor web 21](#_Toc197724218)

[Service pel MySQL 22](#_Toc197724219)

[Service pel PhPMyAdmin 22](#_Toc197724220)

[Persistent Volume Claim 23](#_Toc197724221)

[Desplegar aplicacions a Kubernetes 25](#_Toc197724222)

[Comprovar l'estat dels pods i serveis 26](#_Toc197724223)

Introducció

Aquest projecte té como objectiu implementar un desplegament de microserveis amb orquestradors utilitzant Docker Swarm i Kubernetes dins de l’entorn de Proxmox.

**Objectius**

* Implementar un entorn multi-contenidor amb Docker Compose.
* Configurar i desplegar serveis amb Docker Swarm.
* Aplicar mesures de seguretat en un entorn Swarm.
* Migrar el projecte des de Docker Compose a Kubernetes.
* Comparar avantatges i inconvenients de Docker Swarm i Kubernetes.

Configuracions i desplegament amb Docker Compose (Fase 1)

Introducció a Docker Compose

Docker Compose és una eina de l'ecosistema Docker que serveix per definir aplicacions multicontenidor en un fitxer de text anomenat **docker-compose.yml**. Dins aquest fitxer podeu configurar pràcticament les mateixes propietats per als contenidors.

**Característiques:**

* Múltiples entorns aïllats en un sol host.
* Conservació de les dades dels contenidors a través de les diferents execucions.
* Regeneració de contenidors que hagin patit modificacions.
* Configuració de variables i composició de diferents entorns de treball.
* Organització del treball perquè diversos contenidors funcionin junts.

Instal·lació i configuració de Docker Compose

De primeres hem preparat el nostre entorn per fer ús de Docker Compose, hem actualitzat els paquets de la nostre màquina i hem instal·lat el docker.io.

Texto

El contenido generado por IA puede ser incorrecto.

Un cop instal·lat activem el servei docker i verifiquem que està actiu.

Texto

El contenido generado por IA puede ser incorrecto.

Després ens descarreguem el docker-compose vía GitHub per tenir instal·lat la última versió.

Pantalla de computadora con números

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Un cop tot instal·lat verifiquem que el Docker Compose i el Dcoker están actius i funcionen.

Texto

El contenido generado por IA puede ser incorrecto.

Després de tota la instal·lació hem afegit l’usuari creat de la nostre màquina al grup Docker per evitar d’utilitzar sudo mentre estem treballant.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Creació d’un entorn multi-contenidor

Un entorn multi-contenidor permet executar diferents serveis en contenidors separats que es comuniquen entre ells.

**Apache 🡪** Gestiona peticions web.

**MySQL 🡪** Base de dades.

**PHP** 🡪 permet executar codi dinàmic.

**phpMyAdmin** 🡪 Gestió de la base de dades via interfície web.

Aquesta es l’estructura de directoris de la nostre máquina virtual, aquí guardem el fitxer Docker-compose, dos directoris més on guardem els phps de les pàgines i l’altre on es crea la taula usuaris en la base de dades.

Texto

El contenido generado por IA puede ser incorrecto.

Dins del directori lamp-project hem creat el fitxer docker-compose.yml on hem definit un entorn LAMP amb tres serveis: Apache amb PHP, una base de dades MySQL i el phpMyAdmin.

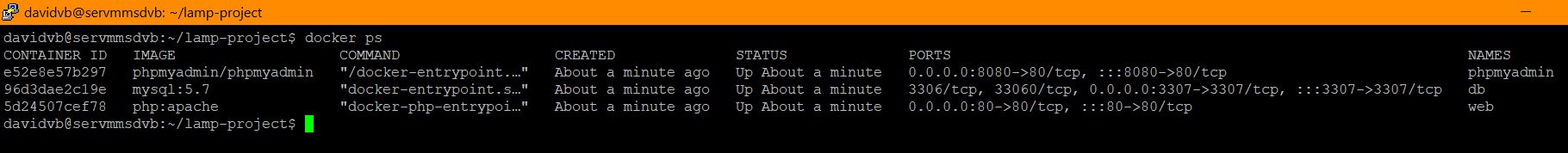
Texto

El contenido generado por IA puede ser incorrecto.

Un cop hem creat el fitxer docker-compose.yml despleguem el serveis i executem els contenidors en segon pla.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

****Després comprovem el correcte funcionament dels contenidors, on tenim el MySQL en el port 3307, Apache està mapejant correctament al port 80 i el PhPMyAdmin és accessible pel port 8080.

Dins del contenidor del MySQL hem creat la taula usuaris dins de la base de dades projecte, en aquesta taula es guardarà totes les dades que es registrin en el formulari que hem creat.

**Texto

El contenido generado por IA puede ser incorrecto.**

Després verifiquem que tenim el php instal·lat dins del contenidor web.

**Texto

El contenido generado por IA puede ser incorrecto.**

Ara hem cercat per Google la IP amb el port corresponent del contenidor web i ens apareix el formulari que hem fet.

Comprovem que el formulari funciona afegint les dades que ens demana.

**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

**Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.**

Polsant el botó de **Veure Usuaris Registrats** es pot veure l’usuari que hem afegit anteriorment amb la imatge.

**Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.**

Aquestes mateixes dades s’han afegit automàticament en la nostre base de dades de usuaris.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

Després podem iniciar sessió amb els usuaris que creem en el formulari de registre.

**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

Un cop iniciem sessió amb un usuari ens mostra les seves dades registrades i diferents botons funcionals.

**Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.**

Per últim tenim un xat on podem xatejar en temps real amb els diferents usuaris creats.

**Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.**

Mentres anem xatejant els missatges es guarden dins de la base de dades de missatges\_chat.

**Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.**

Orquestració i desplegament amb Docker Swarm (Fase 2)

Introducció a Docker Swarm

Docker Swarm és una eina que permet executar els contenidors en una granja de nodes, això implica un o diversos balancejadors de càrrega implementats en un o diversos nodes mestres i els nodes que presten el servei, implementats en nodes treballadors.

Avantatges

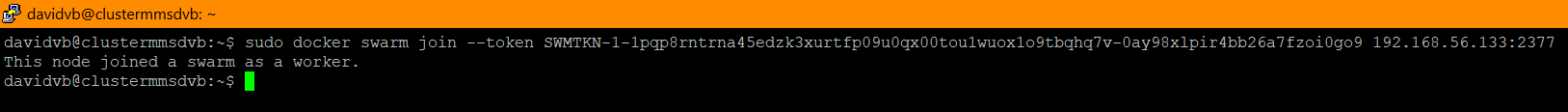
* Permet augmentar i reduir el nombre de rèpliques d’un servei.
* En el cas que un node caigui de forma automàtica els serveis es redistribueixen.
* Es pot integrar molt senzill directament amb Docker.

Configuració d’un clúster Swarm

Iniciem el swarm especificant la nostre IP afegint el nom de la interfície de xarxa perquè d’aquesta forma si canviem la IP no fa falta tornar a executar la comanda.

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Després de crear el clúster, en una altre màquina hem executat el token generat per afegir més nodes al clúster.

Aquí verifiquem que s’ha afegit el node correctament dins del clúster, on es pot veure el hostname de la màquina afegida i el seu status que en aquest cas està actiu.

Texto

El contenido generado por IA puede ser incorrecto.

Migració del projecte de Docker Compose a Docker Swarm

Pel desplegament dels serveis hem creat el fitxer docker-stack.yml dins del directori lamp-project.

Definim un entorn LAMP **(LINUX, Apache, MySQL i PHP).**

Texto

El contenido generado por IA puede ser incorrecto.Texto

El contenido generado por IA puede ser incorrecto.

El servei web utilitza la imatge oficial de PHP, definim una variable d’entorn per configurar el nom del nostre servidor, tenim un **deploy** on tenim **dues instàncies** i una condició si el servei falla es reiniciarà automàticament el Docker Swarm.

Utilitza el port **80** del contenidor en el port 80 de l’amfitrió que permet l’accés a la web.

Mapeja el directori  **./html** de l’amfitrió **/var/www/html** dins del contenidor on permet que els arxius PHP es serveixin des de l’amfitrió.

Texto

El contenido generado por IA puede ser incorrecto.

El servei db utilitza l’última versió del MySQL, definim la configuració de MySQL on afegim la password del root,nom de la base de dades, usuari de la bd i la password de l’usuari, tenim un deploy on tenim una instància i una condició si el servei falla es reiniciarà automàticament.

Utilitza el port 80 del contenidor en el port 80 de l’amfitrió que permet l’accés a la web.

Mapeja el volum **db\_data** al directori **/var/lib/mysql** dins del contenidor per a **persistència de dades.**

Texto

El contenido generado por IA puede ser incorrecto.

El phpmyadmin mapeja el **./phpmyadmin-config** a **/etc/phpmyadmin** dins del contenidor, d’aquesta forma permet des de l’amfitrió modificar configuracions de phpMyAdmin.

El directori /var/lib/mysql l’agafa del mateix docker

**Texto

El contenido generado por IA puede ser incorrecto.**

Desplegar el serveis definits al docker-stack.yml

Texto

El contenido generado por IA puede ser incorrecto.

Un cop desplegat els serveis revisem els serveis desplegats en el clúster de Docker Swarm. Tenim 3 serveis actius db, phpmyadmin i web, en cada servei està configurat en mode replicat.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Seguretat a Docker Swarm (Fase 3)

Tècniques de Seguretat

Primer hem implementat la autenticació basada en certificats d’aquesta forma evitem els accessos no desitjats i permet que només els nodes autoritzats poden unir-se al Docker Swarm.

Texto

El contenido generado por IA puede ser incorrecto.

Xifratge de Comunicacions amb TLS

**Gestió de Secrets**

Dins del directori /opt hem creat l’arxiu secret.txt que permet emmagatzemar claus d’API, contrasenyes i altres dades de forma segura.

Hem decidit guardar-lo en aquest directori i no en /lamp-project per separar-lo i guardar-lo en lloc més segur i amb permisos per evitar que qualsevol usuari pugui accedir.

Texto

El contenido generado por IA puede ser incorrecto.

Després hem creat un secret que s’utilitza pel servei db pero proteger l’accés a la nostre base de dades, així evitem exposar el codi del projecte i altres fitxers visibles.

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Orquestració amb Kubernetes (Fase 4)

Introducció

Kubernetes és una plataforma d'orquestració de contenidors de codi obert per gestionar, automatitzar i escalar aplicacions a contenidors.

Creació i gestió de pods i serveis

Deployment pel servidor web

En el fitxer web-deployment.yml configurem i despleguem una aplicació amb dues repliques del contenidor on executem el PHP amb apache. Especifiquem que el volum html-volume estigui vinculat a la carpeta dins del sistema d’arxius del node al contenidor, d’aquesta forma permet que els arxius dins d ‘aqueta carpeta siguin accessibles al contenidor. Escolta el port 80 per mostrar el contingut HTML.

Texto

El contenido generado por IA puede ser incorrecto.

Deployment pel MySQL

En el fitxer db-deployment.yml configurem i despleguem una aplicació amb una replica del contenidor de MySQL. Aquest Deployment utilitza la imatge de mysql on configurem les diferents credencials com nom de bd, password i usuari, també monta un volum al directori /var/lib/mysql per evitar la pèrdua de dades quan el pod es reincia.

Texto

El contenido generado por IA puede ser incorrecto.

Deployment PhPMyAdmin

En el fitxer phpmyadmin-deployment.yml configurem i despleguem un contenidor amb phpmyadmin amb una replica. Aquest Deployment utilitza la imatge de phpmyadmin, després definim amb la variable PMA\_HOST que es connecti al servei MySQL i escolta el port 80 per poder accedir via web.

Texto

El contenido generado por IA puede ser incorrecto.

Service per exposar el servidor web

En el fitxer web-service.yml configurem un servei de tipus NodePort anomenat web-service per exposar una aplicació web que escolta pel port 80. Aquest servei selecciona els pods que tenen l’etiqueta app: web, i redirigeix el trànsit cap al port 80 dels contenidors. A més, assignem el nodePort 31604 per poder accedir a l’aplicació des de fora del clúster a través de qualsevol node de Kubernetes.

Texto

El contenido generado por IA puede ser incorrecto.

Service pel MySQL

En el fitxer db-service.yml definim un servei anomenat db-service per donar accés a la base de dades MySQL. Aquest servei selecciona els pods amb l’etiqueta app: db i redirigeix el trànsit cap al port 3306, que és el port per defecte de MySQL. A més, especifiquem clusterIP: None per crear un Headless Service, permetent una comunicació directa amb els pods i facilitant la descoberta del servei dins del clúster, ideal quan utilitzem volums persistents o StatefulSets.

**Texto

El contenido generado por IA puede ser incorrecto.**

Service pel PhPMyAdmin

En el fitxer phpmyadmin-service.yml definim un servei anomenat phpmyadmin-service que permet accedir a la interfície web de phpMyAdmin. Aquest servei selecciona els pods amb l’etiqueta app: phpmyadmin i exposa el port 8080 del clúster, redirigint-lo internament al port 80 del contenidor. El tipus de servei és LoadBalancer, cosa que permet que aquest servei pugui ser accessible des de fora del clúster, proporcionant una IP externa.

**Texto

El contenido generado por IA puede ser incorrecto.**

Persistent Volume Claim

En el fitxer mysql-pvc.yml definim un **PersistentVolumeClaim** anomenat mysql-pv-claim, que sol·licita un espai d’emmagatzematge de **1 GiB**. Aquest volum permet al contenidor de la base de dades **MySQL** desar les dades de forma persistent, de manera que es conserven encara que el pod es reiniciï o s’elimini. L’**accessMode** ReadWriteOnce indica que el volum només pot ser muntat en mode lectura-escriptura per un únic node alhora.

**Texto

El contenido generado por IA puede ser incorrecto.**

En el fitxer namespace.yml definim un Namespace anomenat lamp-network. Aquest namespace ens permet organitzar i separar els recursos del clúster relacionats amb el projecte LAMP, com ara deployments, serveis i volums. L’ús de namespaces facilita la gestió i el control dels recursos, especialment quan es treballa amb diversos entorns o equips dins del mateix clúster de Kubernetes.

**Texto

El contenido generado por IA puede ser incorrecto.**

Un cop hem creat tots els fitxers service y deployments

**Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.**

Ara instal·lem el minikube a /usr/local/bin/ per poder executar el clúster de Kubernetes.

**Texto

El contenido generado por IA puede ser incorrecto.**

Desplegar aplicacions a Kubernetes

Ara fem el desplegament dels diferents recursos del nostre entorno n hm creat el deployment bd, PersistentVolumeClaim, name sapce, deployments i servei phpmyadmin i per últim el Deployment i servei del servei web.

**Texto

El contenido generado por IA puede ser incorrecto.**

Comprovar l'estat dels pods i serveis

Per últim comprovem que tots els pods estan operatius dins del clúster amb 1 rèplica activa i sense errors.

Després comprovem l’estat dels serveis on podem veure els ports i el tipus de servei que utilicen.

**Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.**