

1. GLPI es una aplicación de gestión de hardware y software que utiliza una base de datos MariaDB. Sigue los pasos indicados para crear un contenedor para la base de datos de MariaDB y un contenedor de GLPI.

```
docker pull diouxx/glpi

docker network create --subnet=172.18.0.0/16 red_glpi

docker run --name mariadb --net red_glpi --ip 172.18.0.100 -e
MARIADB_ROOT_PASSWORD=diouxx -e MARIADB_DATABASE=glpidb -e
MARIADB_USER=glpi_user -e MARIADB_PASSWORD=glpi -d mariadb:10.7

docker run --name glpi --net red_glpi -p 8080:80 -d diouxx/glpi
```

2. Muestra el detalle de la red red_glpi.

```
[
  {
    "Name": "red_glp1",
    "Id": "7edb37deb3866671e2997e8411440efba4f7473d62eb03a96675071f5fcea249",
    "Created": "2024-11-28T11:44:17.495855563+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": {},
      "Config": [
        {
          "Subnet": "172.18.0.0/16"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "92726724990f697dba034300c60789877ec7d1c56b44abf78ee6e7cad9ce1809": {
        "Name": "glpi",
        "EndpointID": "a769d85ac9b09478df36f6952def00507ad7379f98908dfa3e1ba291b5e136de",
        "MacAddress": "02:42:ac:12:00:02",
        "IPv4Address": "172.18.0.2/16",
        "IPv6Address": ""
      },
      "e0cd7e547df0d6262993a0aa5ad50385ed72788031231a725a6f6c2c1355ab73": {
        "Name": "mariadb",
        "EndpointID": "44cf8690f4a1444052f13d9042201ade09f4e75687a11f1a25d3ae960c77dbf9",
        "MacAddress": "02:42:ac:12:00:64",
        "IPv4Address": "172.18.0.100/16",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
alumno@pc-alumno:~$
```

3. Responde a las siguientes preguntas:

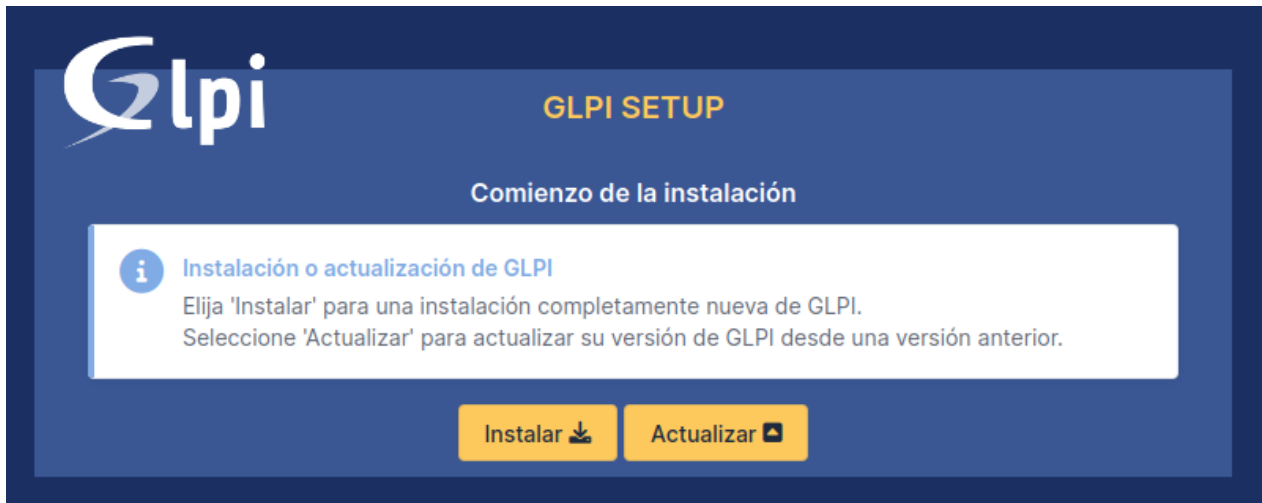
- ¿Cómo se llaman los contenedores que has creado?
Uno se llama glpi y el otro mariadb
- ¿Qué tipo de red se ha creado?
Es una red de tipo bridge
- ¿Se podrán comunicar entre ellos?
Al estar en una red bridge los contenedores si se pueden comunicar entre si
- ¿Qué IP se ha asignado a cada contenedor?
Al contenedor glpi = 172.18.0.2/16 y a mariadb = 172.18.0.100/16
- ¿Las IPs se han asignado de forma manual o automática?
Se han asignados de forma manual a mariadb
- ¿Para qué se usa la opción -e?

Esta opción le permite especificar variables de entorno arbitrarias que están disponibles para el proceso que se iniciará dentro del contenedor.

- g. ¿A qué puerto hay que conectarse para acceder al servidor de GLPI?
Hay que conectarse al puerto 8080:80
4. Accede a la dirección <http://localhost:8080> e incluye capturas de todos los pasos que has tenido que seguir hasta tener el servidor disponible. Fíjate bien en cómo se ha creado el contenedor de la base de datos (**mariadb**) para extraer los datos que te van a solicitar.



The image displays two screenshots of the GLPI Setup interface. The top screenshot shows the 'GLPI SETUP' screen with the 'Elija su idioma' (Choose your language) section. A dropdown menu is open, showing 'Español (España)' (Spanish (Spain)) as the selected language. Below the dropdown is a yellow button labeled 'Correcto >' (Correct >). The bottom screenshot shows the 'GLPI SETUP' screen with the 'Licencia' (License) section. It displays the 'GNU GENERAL PUBLIC LICENSE Version 3, 29 June 2007' text. Below the license text is a blue button labeled 'Continuar >' (Continue >). A link at the bottom of the license section reads 'También están disponibles traducciones no oficiales' (Unofficial translations are also available).





GLPI SETUP

Paso 0

Verificar la compatibilidad de su entorno con la ejecución de GLPI

PRUEBA REALIZADA	RESULTADOS
Requerido Intérprete PHP	✓
Requerido Configuración de sesiones	✓
Requerido Memoria asignada	✓
Requerido mysql extensión	✓
Requerido Extensiones del núcleo de PHP	✓
Requerido curl extensión <i>Necesario para el acceso remoto a los recursos (solicitudes de agentes de inventario, mercado, fuentes RSS, ...).</i>	✓
Requerido gd extensión <i>Requerido para el manejo de imágenes.</i>	✓
Requerido intl extensión <i>Requerido para la internacionalización.</i>	✓
Requerido zlib extensión <i>Necesario para el manejo de comunicación comprimida con agentes de inventario, instalación de paquetes gzip desde el mercado y generación de PDF.</i>	✓
Requerido Tamaño de la constante Sodium ChaCha20-Poly1305 <i>Activar el uso del cifrado ChaCha20-Poly1305 requerido por GLPI. Esto lo proporciona libsodium 1.0.12 y versiones posteriores.</i>	✓
Requerido Permisos para archivos de registro	✓
Requerido Permissions for GLPI data directories	✓
Seguridad PHP maintained version <i>Se debería usar una versión de PHP con mantenimiento, para beneficiarse de parches de seguridad y corrección de errores.</i>	✓
Seguridad Configuración segura del directorio raíz web <i>El directorio raíz del servidor web debe ser '/var/www/html/glpi/public' para garantizar que no se pueda acceder a los archivos no públicos.</i>	✓
Seguridad Configuración de seguridad para sesiones <i>Asegurar de que se aplica la seguridad en las cookies de sesión.</i>	✓
Sugerido Tamaño máximo de un entero de PHP <i>Se requiere soporte de enteros de 64 bits para operaciones relacionadas con direcciones IP (inventario de red, filtrado de IP de clientes API, ...).</i>	✓
Sugerido exif extensión <i>Mejore la seguridad en la validación de imágenes.</i>	✓
Sugerido ldap extensión <i>Activar el uso de la autenticación a través del servidor LDAP remoto.</i>	✓
Sugerido openssl extensión <i>Activar el envío de correo electrónico mediante SSL/TLS.</i>	✓
Sugerido Extensiones PHP para el mercado <i>Habilitar la compatibilidad con los formatos de paquetes más comunes en el mercado.</i>	✓
Sugerido Zend OPcache extensión <i>Mejore el rendimiento del motor PHP.</i>	✓
Sugerido Extensiones emuladas de PHP <i>Mejora ligeramente los rendimientos.</i>	✓
Sugerido Permisos para el directorio del mercado <i>Activar la instalación de complementos desde el mercado.</i>	✓

Continuar >



5. Abre la herramienta GLPI e incluye algunas capturas de su funcionamiento.

6. Lista las imágenes.

```
alumno@pc-alumno:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
httpd         latest   dad6ca1caf78   4 months ago   148MB
diouxx/glpi   latest   21cf900de1c9   5 months ago   320MB
mariadb       10.7     895b6c8829c3   19 months ago  396MB
alumno@pc-alumno:~$
```

7. Lista los contenedores.

```
alumno@pc-alumno:~$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
92726724990f   diouxx/glpi   "/opt/glpi-start.sh"    About an hour ago   Up About an hour   443/tcp, 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp   glpi
e0cd7e547df0   mariadb:10.7   "docker-entrypoint.s..." About an hour ago   Up About an hour   3306/tcp                                   mariadb
4a61e1fb53ee   httpd         "httpd-foreground"      47 hours ago       Up About an hour   0.0.0.0:88->80/tcp, [::]:88->80/tcp               servidor_web_1
alumno@pc-alumno:~$
```

8. Inicia el contenedor de Apache httpd.

```
alumno@pc-alumno:~$ docker start servidor_web_1
servidor_web_1
alumno@pc-alumno:~$
```

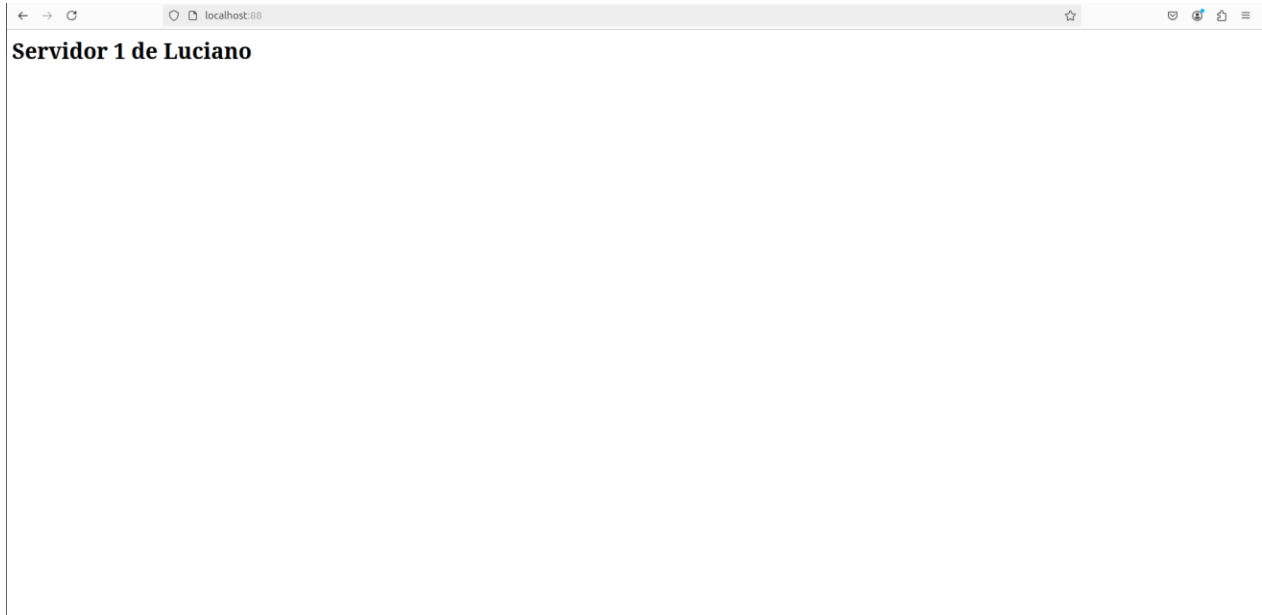
9. Accede al shell de este contenedor y ejecuta el siguiente comando sustituyendo *nombre* por tu nombre:

```
echo "<html><body><h1>Servidor 1 de nombre</h1></body></html>" > htdocs/index.html
```

A continuación, sal de este shell con el comando `exit`.

```
alumno@pc-alumno:~$ docker exec -it servidor_web_1 bash
root@4a61e1fb53ee:/usr/local/apache2# echo "<html><body><h1>Servidor 1 de Luciano</h1></body></html>" > httdocs/index.html
root@4a61e1fb53ee:/usr/local/apache2# exit
exit
alumno@pc-alumno:~$
```

10. Accede a tu servidor web e incluye una captura de la página de inicio.



11. Crea otro contenedor de Apache httpd llamado **servidor_web_2** (tendrá que escuchar en un puerto distinto).

```
alumno@pc-alumno:~$ docker run --name servidor_web_2 -p 8085:80 -d httpd
1bc242866edcb3d6421dfe81fc829f1148cf33d3f0fd503435f7e5ed37e9e4d1
alumno@pc-alumno:~$ docker exec -it servidor_web_2 bash
root@1bc242866edc:/usr/local/apache2# echo "<html><body><h1>Servidor 2 de Luciano</h1></body></html>" > httdocs/index.html
root@1bc242866edc:/usr/local/apache2# exit
exit
alumno@pc-alumno:~$
```

12. Accede al shell de este contenedor y ejecuta el siguiente comando sustituyendo *nombre* por tu nombre:

```
echo "<html><body><h1>Servidor 2 de nombre</h1></body></html>" > httdocs/index.html
```

A continuación, sal de este shell con el comando `exit`.

13. Accede a este segundo servidor web e incluye una captura de la página de inicio.

← → ↻ localhost:8085 ☆ 📄 🏠 ☰

Servidor 2 de Luciano

14. Crea otro contenedor de Apache httpd llamado **servidor_web_3** sin mapear el puerto 80 y usando el controlador de red **host**.
15. Accede al shell de este contenedor y ejecuta el siguiente comando sustituyendo *nombre* por tu nombre:

```
echo "<html><body><h1>Servidor 3 de nombre</h1></body></html>" > htdocs/index.html
```

A continuación, sal de este shell con el comando `exit`.

```
alumno@pc-alumno:~$ docker run --name servidor_web_3 --network host -d httpd
89a786a8ec54b315f21c2f011dd09e68045884e4612a43e9e2f0968b751781cf
alumno@pc-alumno:~$ docker exec -it servidor_web_3 bash
root@pc-alumno:/usr/local/apache2# echo "<html><body><h1>Servidor 3 de Luciano</h1></body></html>" > htdocs/index.html
root@pc-alumno:/usr/local/apache2# exit
exit
alumno@pc-alumno:~$
```

16. Muestra las redes disponibles.

```
alumno@pc-alumno:~$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
2c8c4750fab3        bridge             bridge              local
bbb39390c11b        host               host                local
4953f433901f        none              null                local
7edb37deb386        red_glp            bridge              local
alumno@pc-alumno:~$
```


17. Muestra el detalle de las redes **bridge** y **host**.

```
alumno@pc-alumno:~$ docker network inspect host
[
  {
    "Name": "host",
    "Id": "bbb39390c11bf3c63abc956a472e0318f3985c12d0c677e0dc868807bd140aab",
    "Created": "2024-11-26T13:09:26.024141711+01:00",
    "Scope": "local",
    "Driver": "host",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": null
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "89a786a8ec54b315f21c2f011dd09e68045884e4612a43e9e2f0968b751781cf": {
        "Name": "servidor_web_3",
        "EndpointID": "8703a22a5458471c7264aecdcc183864d2a908afe0190a93c0ecb9a0666efb5",
        "MacAddress": "",
        "IPv4Address": "",
        "IPv6Address": ""
      }
    },
    "Options": {},
    "Labels": {}
  }
]
```

alumno@pc-alumno:~\$

```
alumno@pc-alumno:~$ docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "2c8c4750fab38356a7b44b654ae95869a1026710858f8cdeb8f8f655d93cfe6eb",
    "Created": "2024-11-28T11:25:53.123430811+01:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16",
          "Gateway": "172.17.0.1"
        }
      ]
    },
    "Internal": false,
    "Attachable": false,
    "Ingress": false,
    "ConfigFrom": {
      "Network": ""
    },
    "ConfigOnly": false,
    "Containers": {
      "1bc242866edcb3d6421dfe81fc829f1148cf33d3f0fd503435f7e5ed37e9e4d1": {
        "Name": "servidor_web_2",
        "EndpointID": "252c083e0ddb12286a9316b2d9da0edbb52c01eb0d6c199e0058674a4fa5ea72",
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
      },
      "4a61e1fb53eea20c57b51db10cd855448aaa3dc178d1b676c6091e52333bb586": {
        "Name": "servidor_web_1",
        "EndpointID": "fd500175213adec854e117cc584625b3c93ed54f880ff2efa7f0ea2f89808f0f",
        "MacAddress": "02:42:ac:11:00:02",
        "IPv4Address": "172.17.0.2/16",
        "IPv6Address": ""
      }
    },
    "Options": {
      "com.docker.network.bridge.default_bridge": "true",
      "com.docker.network.bridge.enable_icc": "true",
      "com.docker.network.bridge.enable_ip_masquerade": "true",
      "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
      "com.docker.network.bridge.name": "docker0",
      "com.docker.network.driver.mtu": "1500"
    },
    "Labels": {}
  }
]
alumno@pc-alumno:~$
```

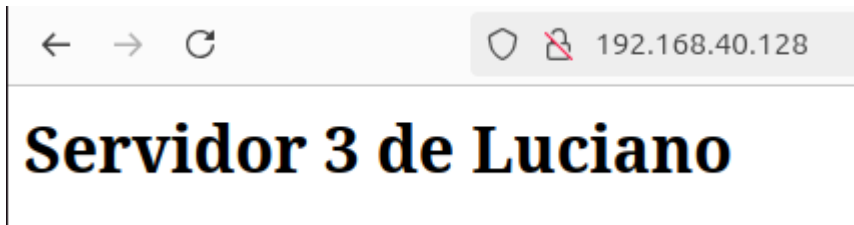
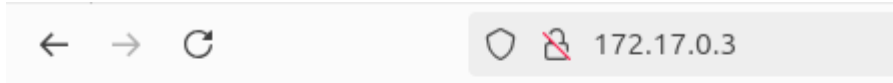
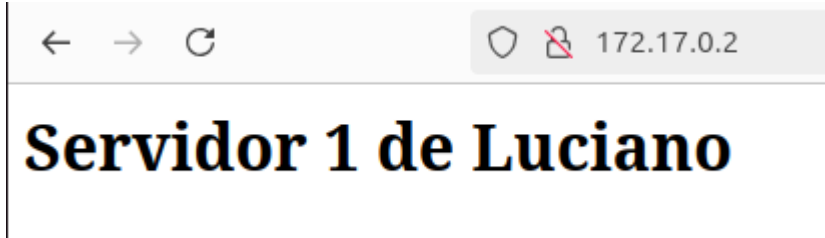
18. ¿Qué dirección IP tienen los servidores 1 y 2? ¿Y cuál tiene el servidor 3?

Server 1 = 172.17.0.2

Server 2 = 172.17.0.3

Server 3 = 192.168.40.128

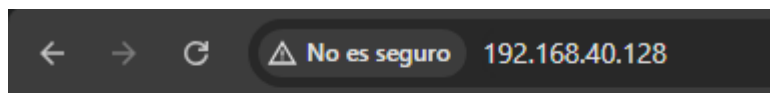
19. Comprueba que se puede acceder a los dos servidores Web que están en la red **bridge** directamente con la IP que se les ha asignado. Por tanto, hay una red entre el host y los dos contenedores y se pueden comunicar entre ellos.



20. ¿Qué diferencia hay con el servidor 3?

La diferencia es que el servidor 3 funciona con la ip del host

21. Has de comprobar ahora el acceso a estos servidores "desde Internet" por lo que te conectarás desde tu equipo. Para ello has de averiguar la dirección IP que tiene la máquina de Ubuntu (por ejemplo, en una ventana de Terminal, con el comando `ip a`).



22. Busca en Docker Hub cómo crear un contenedor de MySQL y créalo añadiendo las opciones `-it`.

```
alumno@pc-alumno:~$ docker run --name some-mysql -e MYSQL_ROOT_PASSWORD=my-secret-pw -it -d mysql
78eaf71308d2e917d1b35a103d467e5987e5300f4fbe4cf3267125be2ec006b7
```

23. Accede al shell del contenedor de MySQL

```
alumno@pc-alumno:~$ docker exec -it some-mysql bash
bash-5.1# mysql --user=root --password
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 9.1.0 MySQL Community Server - GPL
```

24. Ejecuta el siguiente comando del cliente de MySQL para conectar:

```
mysql --user=root --password=contraseña (indica la contraseña que has
configurado al crear el contenedor)
```

25. Una vez dentro del shell de SQL, ejecuta por ejemplo el comando: `show databases;`

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 9.1.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
-> 
```

26. Para los contenedores de glpi y de Apache httpd.

27. Elimina los contenedores de Apache httpd.

```
alumno@pc-alumno:~$ docker stop servidor_web_1 servidor_web_2 servidor_web_3
servidor_web_1
servidor_web_2
servidor_web_3
alumno@pc-alumno:~$ docker rm servidor_web_1 servidor_web_2 servidor_web_3
servidor_web_1
servidor_web_2
servidor_web_3
alumno@pc-alumno:~$
```

28. Lista los contenedores.

```
alumno@pc-alumno:~$ docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
4f535f734a2d   mysql         "docker-entrypoint.s..." 4 days ago    Up 4 days    3306/tcp, 33060/tcp               some-mysql
92726724990f   diouxx/glpi   "/opt/glpi-start.sh"     5 days ago    Up 5 days    443/tcp, 0.0.0.0:8080->80/tcp, [::]:8080->80/tcp glpi
e0cd7e547df0   mariadb:10.7  "docker-entrypoint.s..." 5 days ago    Up 5 days    3306/tcp                          mariadb
alumno@pc-alumno:~$
```

29. Lista las imágenes.

```
alumno@pc-alumno:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql         latest   56a8c14e1404   7 weeks ago    603MB
httpd         latest   dad6ca1caf78   4 months ago    148MB
diouxx/glpi   latest   21cf900de1c9   5 months ago    320MB
mariadb       10.7     895b6c8829c3   19 months ago    396MB
alumno@pc-alumno:~$
```

30. Elimina la imagen de Apache httpd.

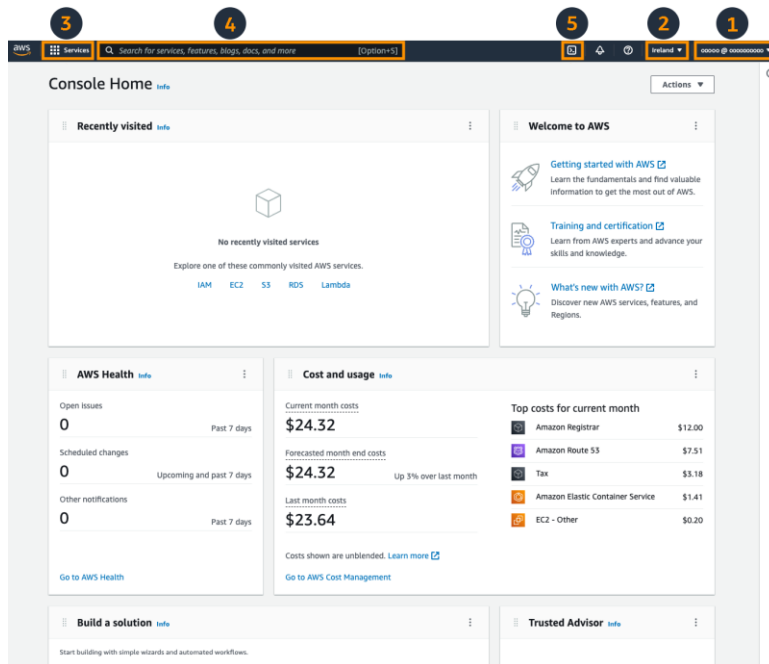
```
alumno@pc-alumno:~$ docker image rm httpd
Untagged: httpd:latest
Untagged: httpd@sha256:6bdbdf5ac16ac3d6ef543a693fd5dfafae2428b4b0cdc52a480166603a069136
Deleted: sha256:dad6ca1caf78e98e22e8e5a406ab4a3427cedfff254c7fcc9dc3b2ca922c642b
Deleted: sha256:092628e738116611e1bda846f338ba9bef1cc2a033dcc436b44b4dacead40078
Deleted: sha256:456ff357a71a1c90c94e4495325203fa902f8382cac26c0a03714dce602dcd99
Deleted: sha256:14d87f17c8b8cdb6546161cf49ff82631b15543e54b678c2dee4f7888456f5f7
Deleted: sha256:565135b3f0ec83dab78e23227d394efd5739080a312269f96b184740e562cb06
Deleted: sha256:559984f97411f386dcbe5f41ffc76ef63f5af393a69c3834317da70eb4d16f48
Deleted: sha256:c3548211b8264f8bfa47a6727043a64f1791b82ac965a284a7ea187e971a95e2
alumno@pc-alumno:~$ docker image ls
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql         latest   56a8c14e1404   7 weeks ago    603MB
diouxx/glpi   latest   21cf900de1c9   5 months ago    320MB
mariadb       10.7     895b6c8829c3   19 months ago    396MB
alumno@pc-alumno:~$
```

31. Busca 2 proveedores de IaaS y 2 de PaaS. Explica en detalle qué servicios ofrecen, el precio y cómo es la consola de administración para el cliente. Indica las referencias consultadas.

Aquí tienes un resumen breve de dos proveedores de IaaS y dos de PaaS:

Proveedores de IaaS:

1. Amazon Web Services (AWS): Ofrece máquinas virtuales (EC2), almacenamiento (S3) y redes virtuales. Su consola permite monitoreo y gestión detallada. El costo de EC2 inicia en \$0.0116 por hora, con opciones de capa gratuita.



2. Google Cloud Platform (GCP): Incluye computación (Compute Engine) y almacenamiento en la nube. Su consola es intuitiva, y los precios empiezan en \$0.0104 por hora con descuentos por uso sostenido.

Práctica 4-3. Creación de contenedores - Cloud computing

All instances > INSTANCE
test-instance: Overview

[EDIT INSTANCE](#) [DELETE INSTANCE](#) [SHOW INFO PANEL](#)

Overview

Name	Test Instance
ID	test-instance
Configuration	us-west1 (Oregon)

Compute capacity ⓘ
Processing units:
1,000
Nodes: 1

CPU utilization (mean)
—

Operations
Read: 0.00/s
Write: 0.00/s

Throughput
Read: —/s
Write: —/s

Total database storage

184 B / 4 TB

Databases

[CREATE DATABASE](#) [REFRESH](#)

Filter Filter databases ⓘ

<input type="checkbox"/>	Name ↑	Dialect ⓘ	CPU utilization	Size	Version retention period ⓘ
<input type="checkbox"/>	example-db	Google Standard SQL	—	92 B (0%)	1 hour
<input type="checkbox"/>	postgresql-db	PostgreSQL	—	0 B	1 hour
<input type="checkbox"/>	restore-db	Google Standard SQL	—	92 B (0%)	1 hour
<input type="checkbox"/>	venues-db	Google Standard SQL	—	0 B	1 hour

Proveedores de PaaS:

1. Heroku: Plataforma para desarrollar en múltiples lenguajes, usando contenedores Dynos. Su consola es sencilla y ofrece planes desde \$7 al mes.

SOLO ENCONTRE ESTO:

hello-world-app-react-jruano · Settings

HEROKU Jump to Favorites, Apps, Pipelines, Spaces...

Personal > hello-world-app-react-jruano [Open app](#) [More](#)

Overview Resources Deploy Metrics Activity Access **Settings**

Name hello-world-app-react-jruano [Edit](#)

Config Vars [Reveal Config Vars](#)

Config vars change the way your app behaves. In addition to creating your own, some addons come with their own.

Info

Region	United States
Stack	heroku-18
Framework	Node.js
Slug Size	46.1 MiB of 500 MiB
Heroku Git URL	https://git.heroku.com/hello-world-app-react-jruano.git

2. AWS Elastic Beanstalk: Soporta aplicaciones en Java, Python, y más, con monitoreo y escalado automático. No tiene costo adicional, solo el de los recursos subyacentes.

Práctica 4-3. Creación de contenedores - Cloud computing

Elastic Beanstalk

Applications

Environments

Change history

Application: GettingStarted

Environment: Gettingstarted-env

Go to environment

Configuration

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Recent environments

Gettingstarted-env

Elastic Beanstalk > Environments > Gettingstarted-env

Gettingstarted-env

Environment overview

Health
Ok

Environment ID
e-lrkuacn9ny

Domain
Gettingstarted-env.eba-w2pdx9as.us-east-1.elasticbeanstalk.com

Application name
GettingStarted

Platform
Node.js 16 running on 64bit Amazon Linux 2/5.7.0

Running version
Sample Application

Events

Health

Logs

Monitoring

Alarms

Managed updates

Tags

Events (38)

Filter events

Time	Type	Details
March 28, 2023 20:01:06 (UTC-4)	INFO	Environment health has transitioned from Info to Ok. Configuration update completed 47 seconds ago and took 14 minutes.
March 28, 2023 20:00:06 (UTC-4)	INFO	Environment update completed successfully.