

DAW

Despliegue de aplicaciones web

Escenarios de despliegue

Juan Jesús Tortajada Cordero

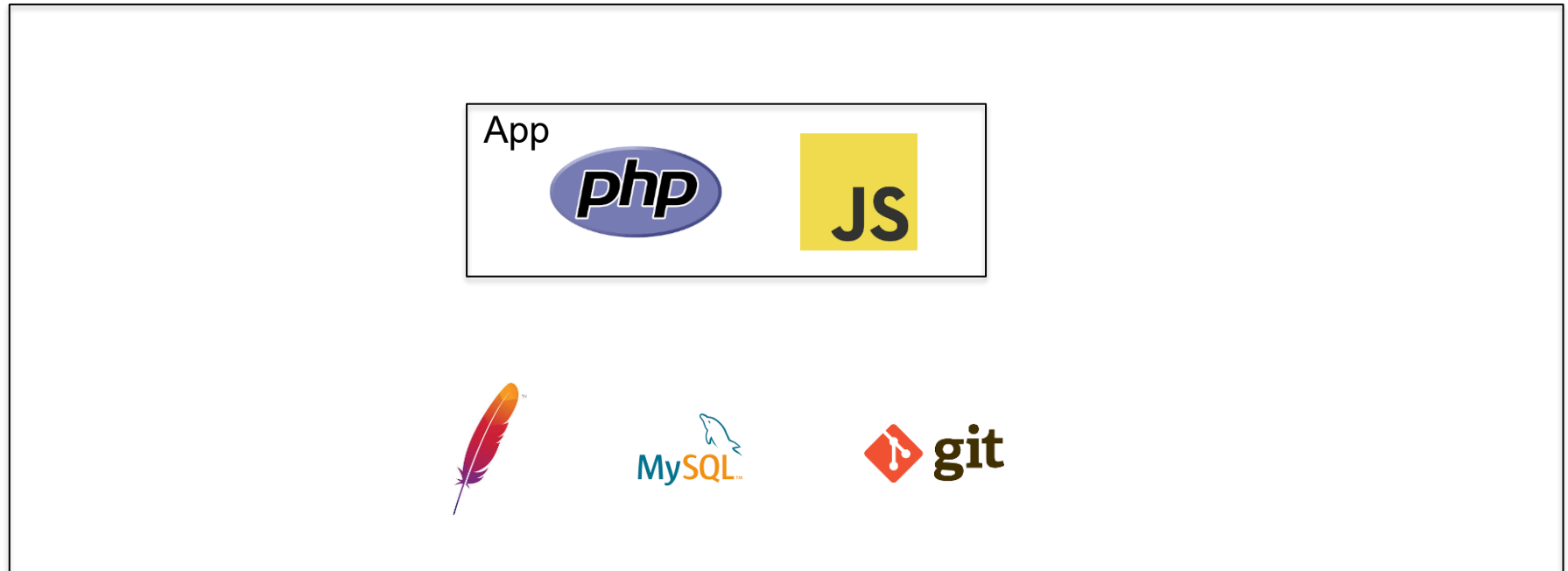
jtortajada@florida-uni.es

- **Escenario 1:** trabajamos solo en local (no hay despliegue).
- **Escenario 2:** trabajo en local, pero con una máquina virtual (MV) simulando un servidor (simulamos un despliegue).
- **Escenario 3:** despliegue manual en una máquina virtual (MV) de AWS.
- **Escenario 4:** despliegue automático en una máquina virtual (MV) de AWS.
- **Escenario 5:** trabajamos solo en local (no hay despliegue) con Docker.
- **Escenario 6:** despliegue manual con Docker.
- **Escenario 7:** despliegue automático con Docker.

Escenario 1: trabajamos solo en local (no hay despliegue).

- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El servidor (Apache/Nginx), la base de datos (MySQL/MongoDB) y el control de versiones están en local.
- Acceso a la aplicación desde un navegador en la URL “localhost”.

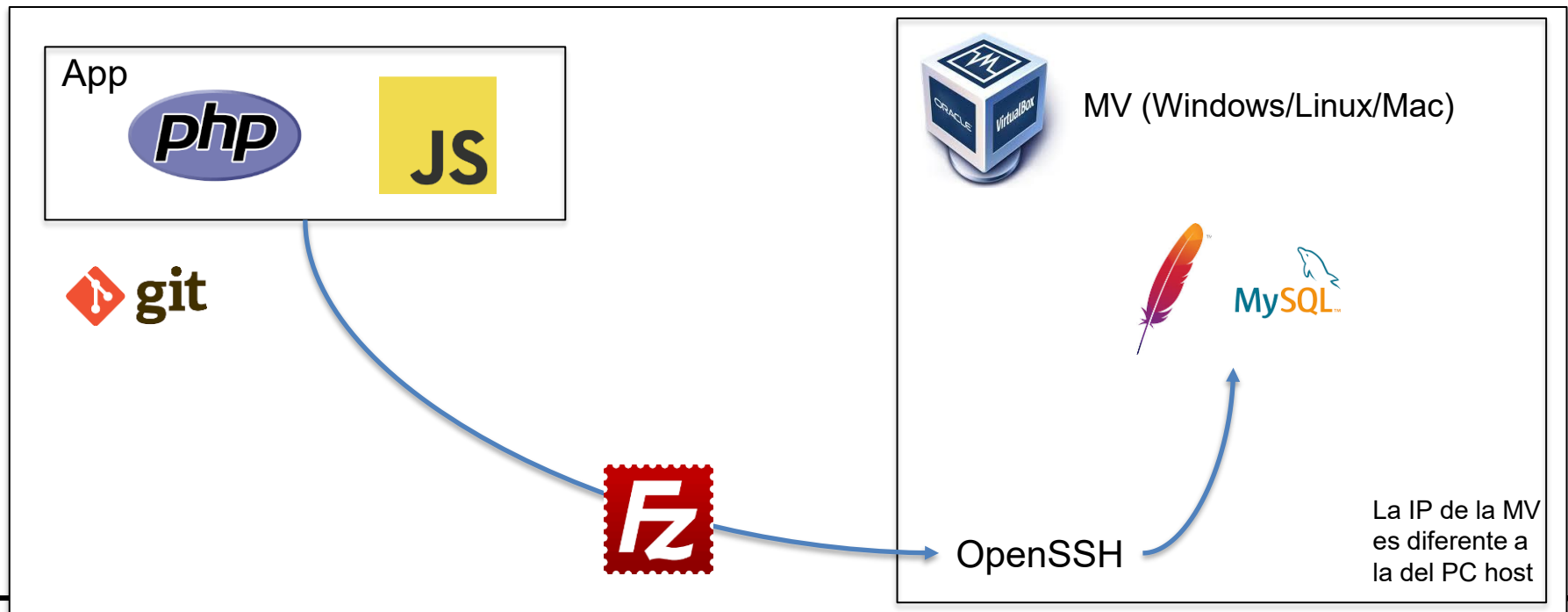
PC host (Windows/Linux/Mac)



Escenario 2: trabajo en local, pero con una máquina virtual (MV) simulando un servidor (simulamos un despliegue).

- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local.
- La MV está en local, pero corre sobre un hipervisor (VirtualBox) con su propio sistema operativo.
- El servidor (Apache/Nginx) y la base de datos (MySQL/MongoDB) están en la MV, que tiene un servidor SSH/SFTP para poder transferir la aplicación vía FileZilla.
- Acceso a la aplicación desde un navegador en la IP de la MV (p.ej. 196.168.0.30).

PC host (Windows/Linux/Mac)



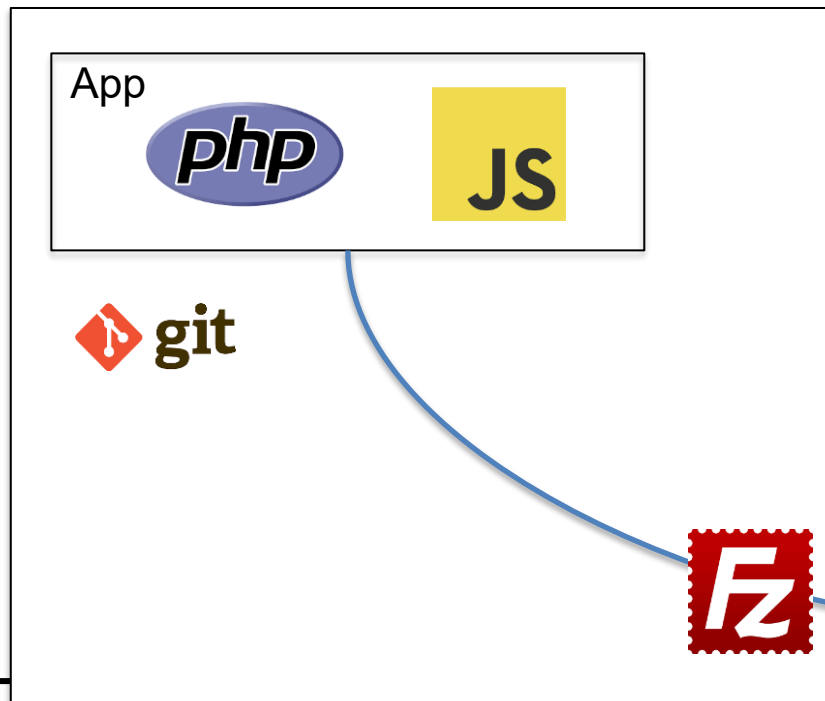
DAW - Despliegue de aplicaciones web

Juan Jesús Tortajada Cordero

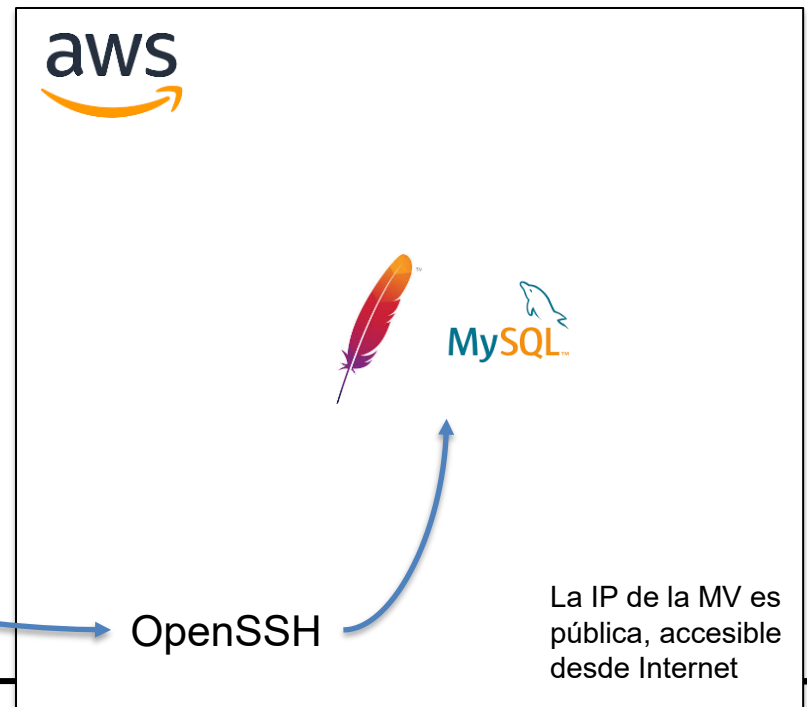
Escenario 3: despliegue manual en una máquina virtual (MV) de AWS.

- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local.
- La MV está en un servidor de AWS, configurado con IP pública.
- El servidor (Apache/Nginx) y la base de datos (MySQL/MongoDB) están en la MV, que tiene un servidor SSH/SFTP para poder transferir la aplicación vía FileZilla.
- Acceso a la aplicación desde un navegador en la IP pública de la MV (p.ej. 43.220.1.85).

PC host (Windows/Linux/Mac)



MV EC2 de AWS (Windows/Linux/Mac)

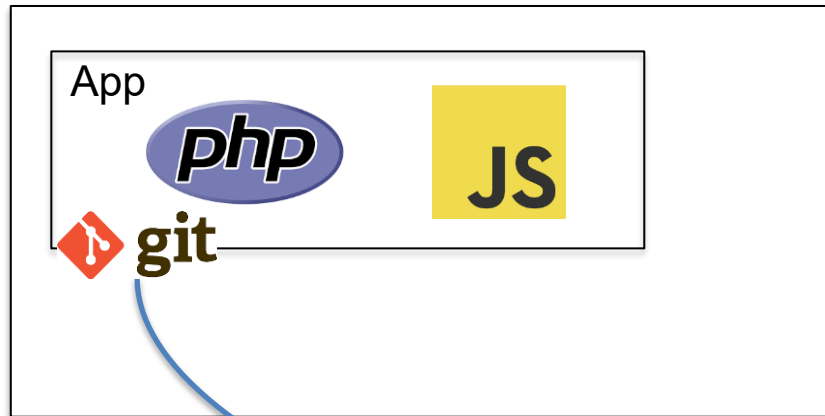


DAW - Despliegue de aplicaciones web

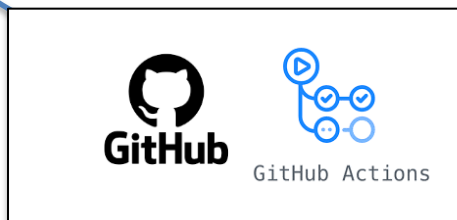
Escenario 4: despliegue automático en una máquina virtual (MV) de AWS.

- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local y está sincronizado con un repositorio de GitHub.
- El repositorio GitHub se configura para despliegue automático (GitHub Actions).
- La MV está en un servidor de AWS, configurado con IP pública.
- El servidor (Apache/Nginx) y la base de datos (MySQL/MongoDB) están en la MV, que tiene un servidor SSH/SFTP para poder transferir la aplicación vía GitHub (GitHub Actions).
- Acceso a la aplicación desde un navegador en la IP pública de la MV (p.ej. 43.220.1.85).

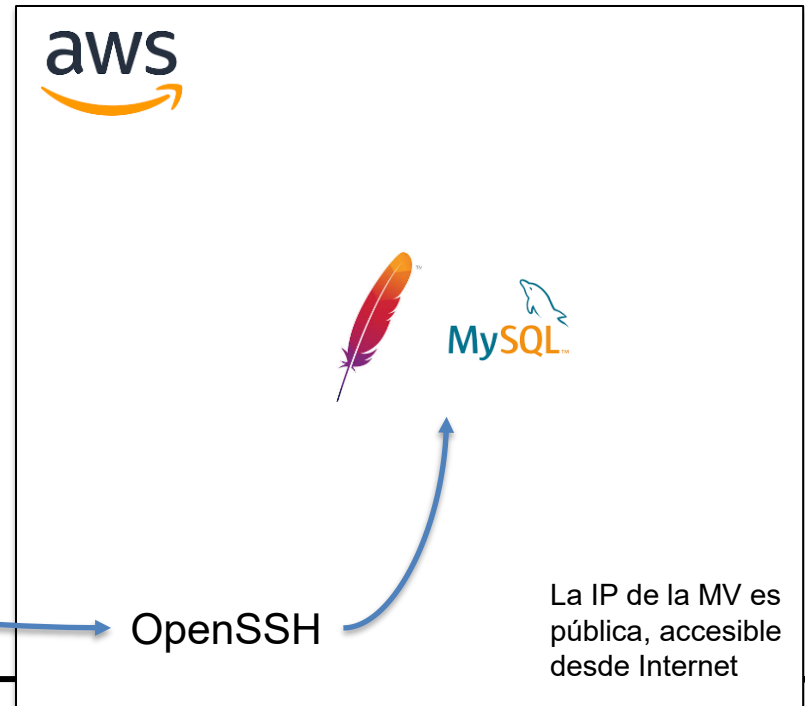
PC host (Windows/Linux/Mac)



Un "git push" activa automáticamente todo el despliegue



MV EC2 de AWS (Windows/Linux/Mac)

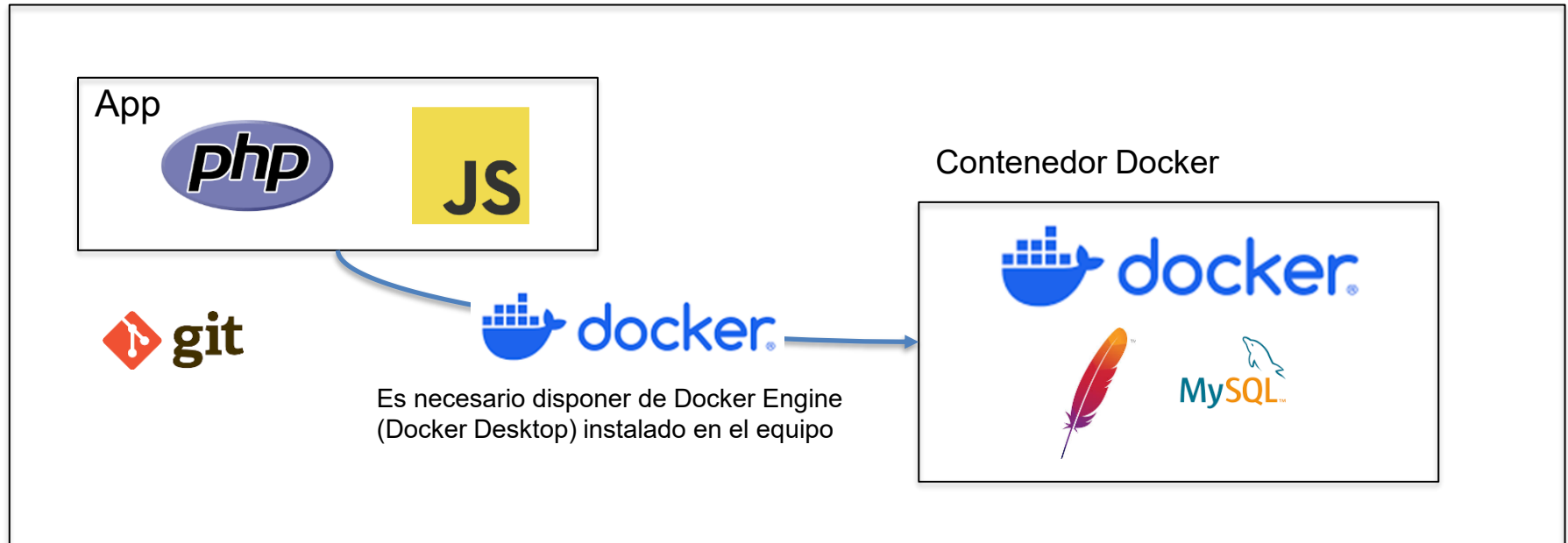


DAW - Despliegue de aplicaciones web

Escenario 5: trabajamos solo en local (no hay despliegue) con Docker.

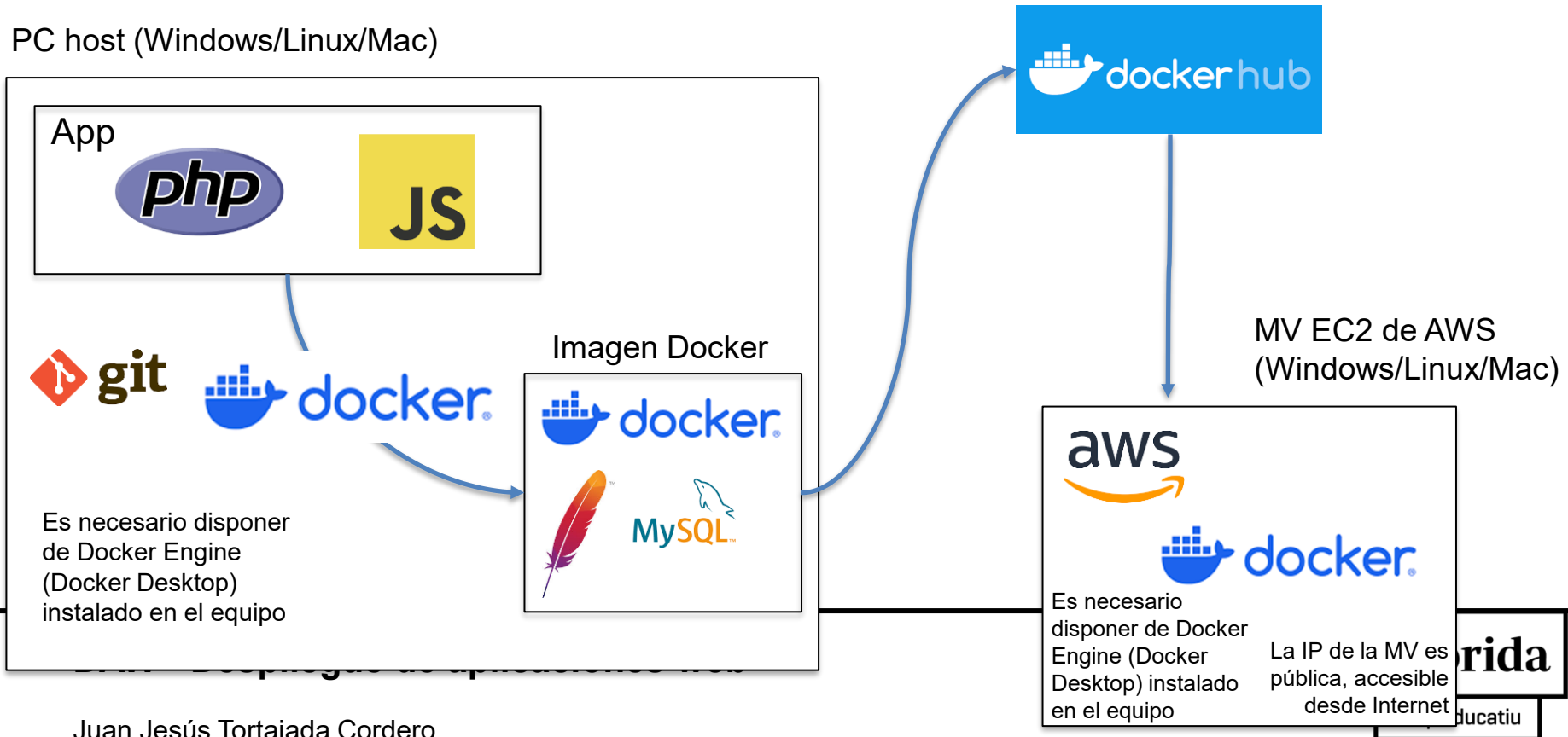
- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local.
- Se definen los servicios necesarios mediante Dockerfile y Docker Compose.
- El servidor (Apache/Nginx) y la base de datos (MySQL/MongoDB) están en Docker.
- La aplicación se copia al Docker (mediante Docker Engine), generando una imagen personalizada y se arranca el contenedor.
- Acceso a la aplicación desde un navegador en la URL “localhost” (p.ej. localhost:8080).

PC host (Windows/Linux/Mac)



Escenario 6: despliegue manual con Docker.

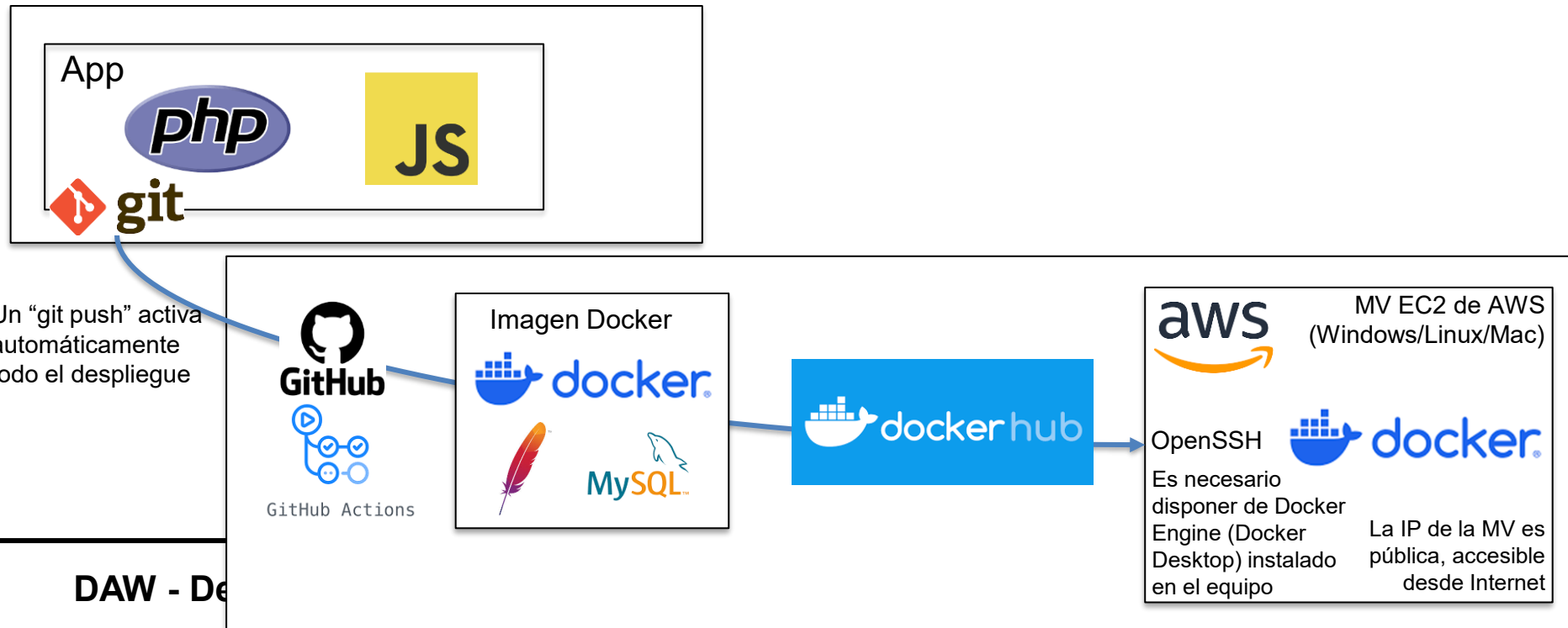
- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local.
- Se definen los servicios necesarios mediante Dockerfile y Docker Compose.
- El servidor (Apache/Nginx) y la base de datos (MySQL/MongoDB) están en Docker.
- La aplicación se copia al Docker (mediante Docker Engine) y se genera una imagen personalizada.
- La imagen Docker se envía a Docker Hub y se descarga y ejecuta como contenedor en cualquier equipo (que tenga instalado Docker Engine previamente).
- Acceso a la aplicación en la IP (pública) del equipo remoto (p.ej. 43.220.1.85:8080).



Escenario 7: despliegue automático con Docker.

- Desarrollamos la aplicación (cliente/servidor, PHP/Javascript) en local.
- El control de versiones está en local y está sincronizado con un repositorio de GitHub.
- El repositorio GitHub se configura para despliegue automático (GitHub Actions).
- GitHub Actions crea la imagen personalizada de Docker con el servidor (Apache/Nginx), la base de datos (MySQL/MongoDB) y la aplicación a partir de uno o varios Dockerfile.
- GitHub Actions envía la imagen a Docker Hub.
- GitHub Actions se conecta vía SSH con la MV, descarga la imagen Docker y la ejecuta.
- Acceso a la aplicación en la IP (pública) del equipo remoto (p.ej. 43.220.1.85:8080).

PC host (Windows/Linux/Mac)



DAW - De