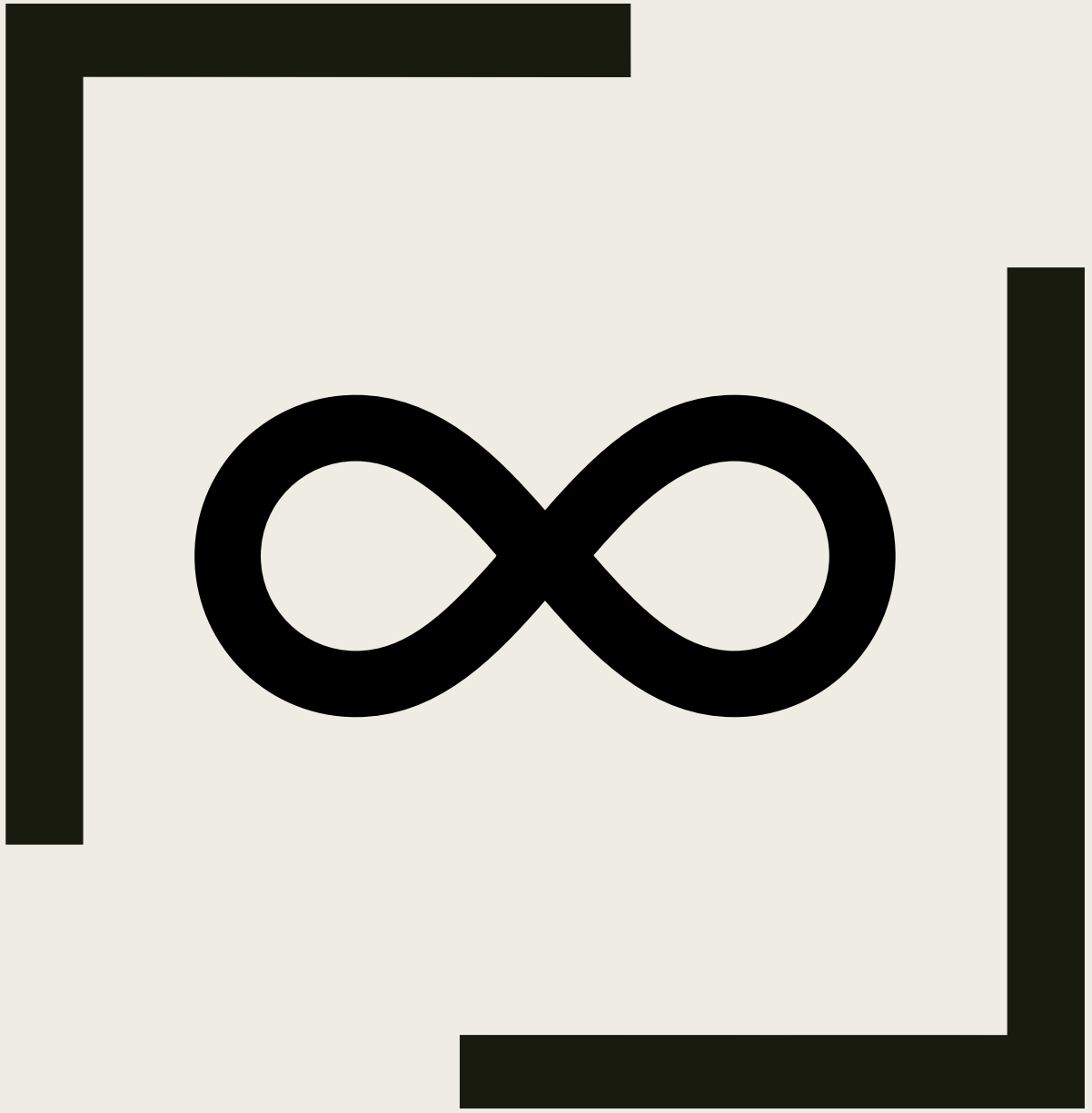
DOCKER

Sesión 4



INCORPORANDO DOCKER AL CICLO DE VIDA SOFTWARE

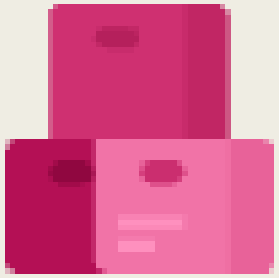
Incorporando Docker al ciclo de vida software

- Hasta ahora, hemos trabajado sobre el repositorio de un servicio, creando un contenedor de esa aplicación para los diferentes entornos.
- **Sin embargo, Docker afecta a todo el ciclo de vida software:**
  - *¿Manejo de dependencias? ¿Credenciales?*
  - *¿Puesta en producción?*
  - *¿Monitorización del servicio?*
  - *¿Gestión de logs?*
  - *¿Versionado del código?*

Hay 3 herramientas que nos pueden ayudar

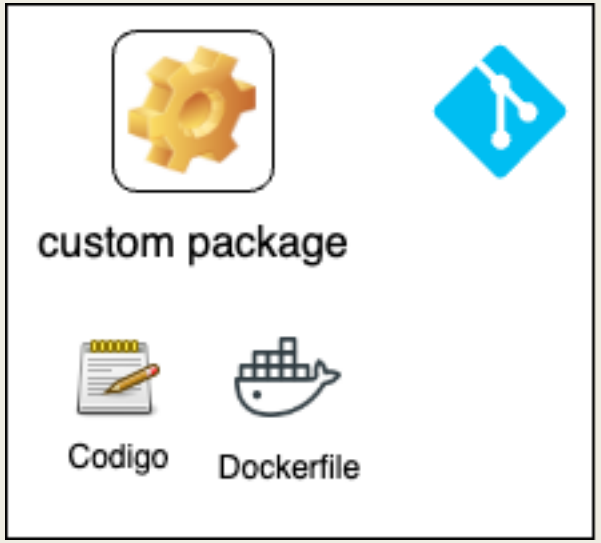
Azure Repos

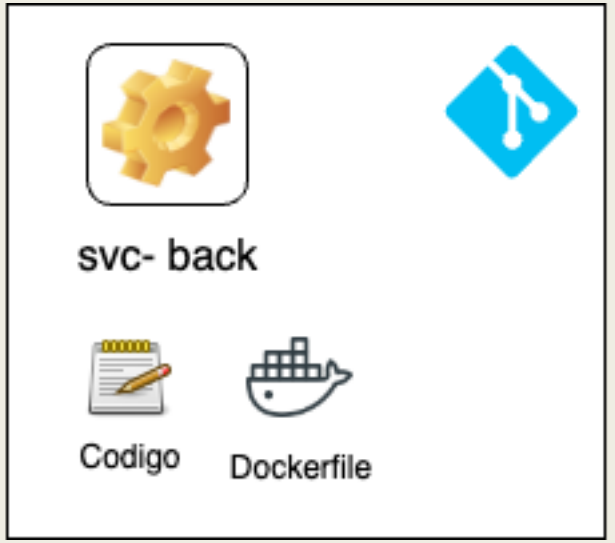
Azure Pipelines

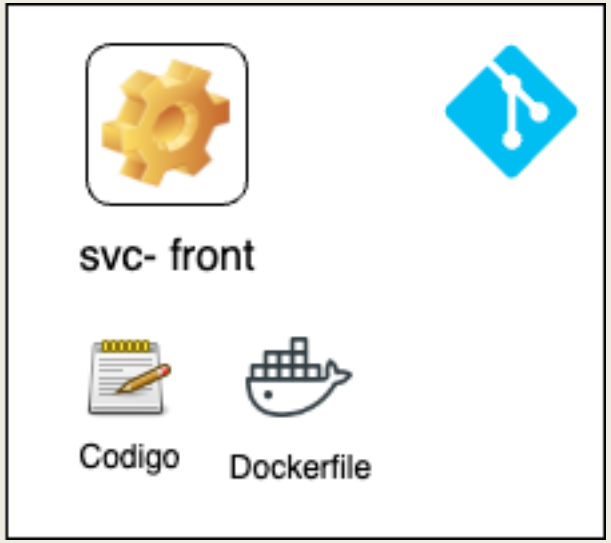


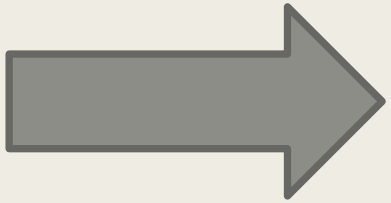
Azure Artifacts

Hasta ahora…



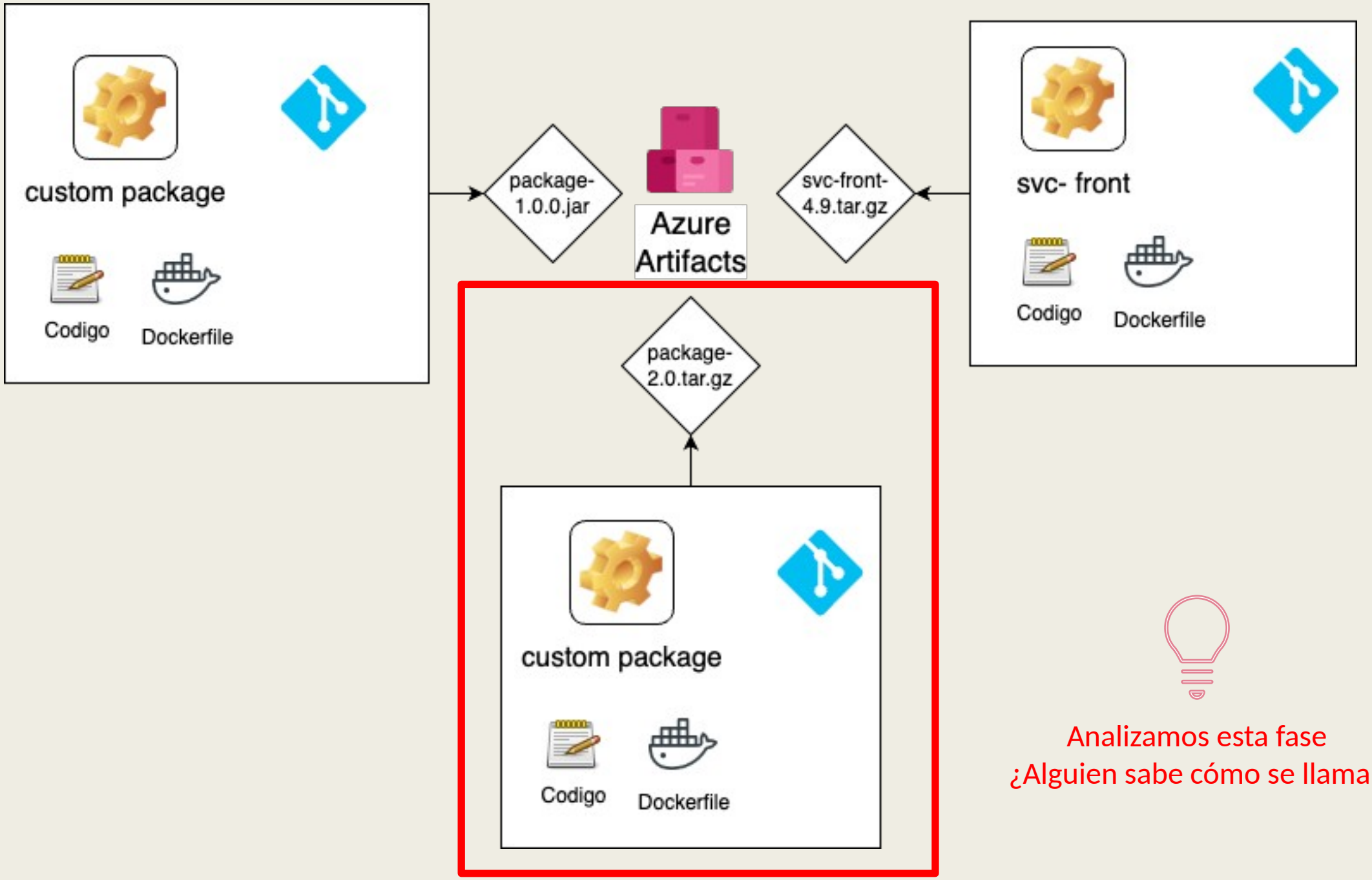






Azure Repos … GitLab, GitHub, etc.

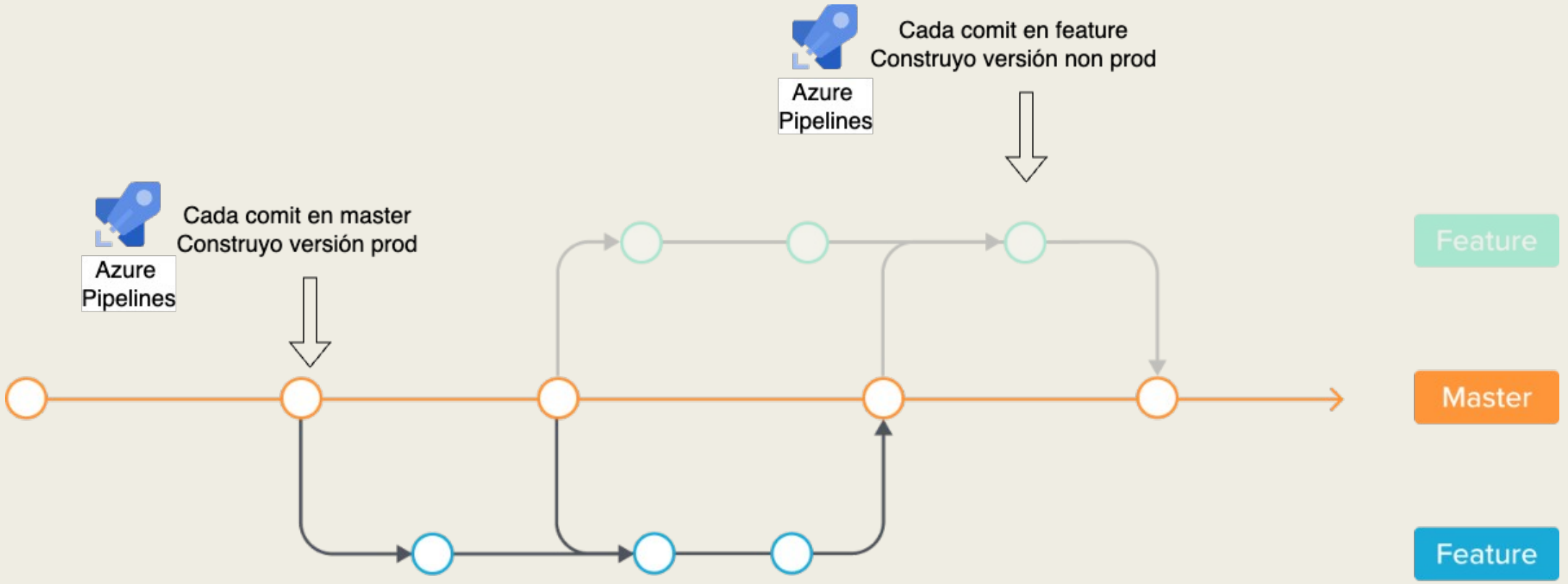
¿Y después?



Analizamos esta fase
¿Alguien sabe cómo se llama?

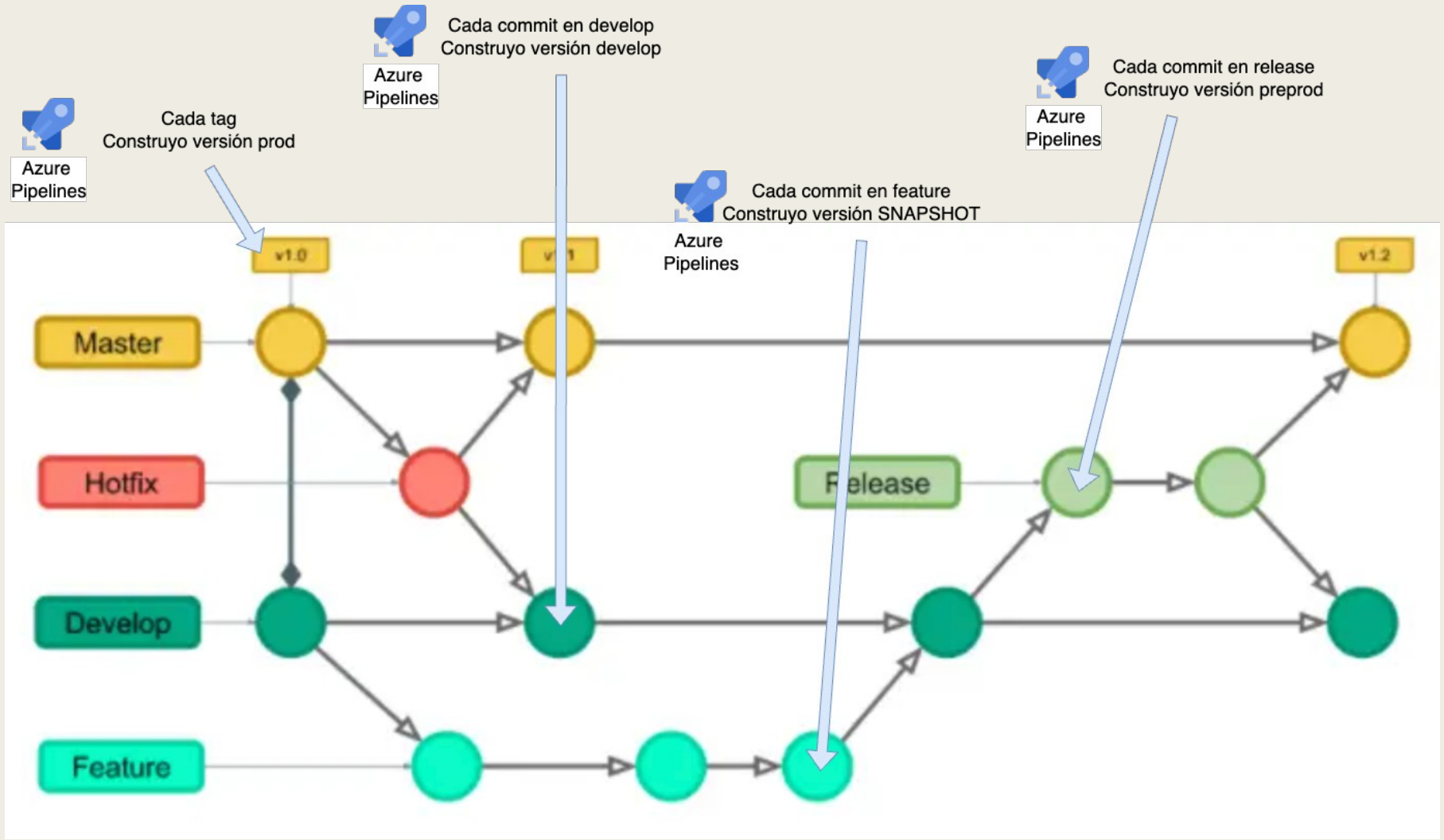
Integración continua (CI)

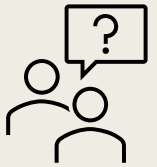
- La forma de implementar la integración continua depende en gran medida del modelo de ramas que adoptemos. Para un modelo de ramas **Trunk Based**:



Integración continua (CI)

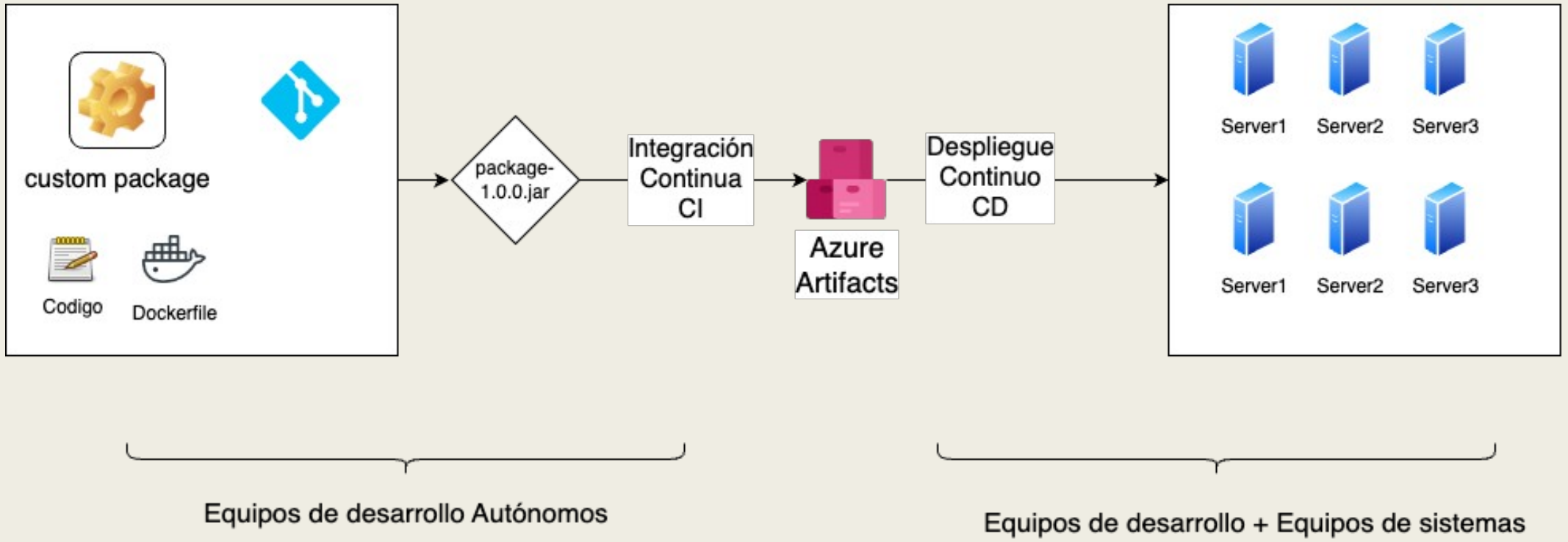
- **Git Flow**



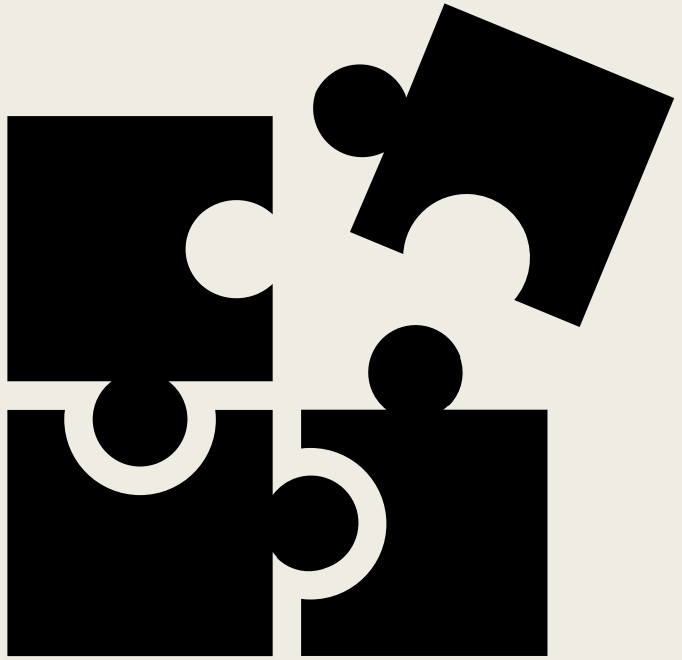


… y después del CI?

Despliegue Continuo (CD)



- ¿Cómo trabajaremos con las dependencias privadas de un repositorio .Net?
- ¿Un contenedor se considera un “artefacto” de mi aplicación?
- ¿Qué usos puede tener usar *tags* “latest”, “estable”, “lts”, etc.?



DOCKER COMPOSE

Docker compose

- **Cliente Docker** para definir y ejecutar aplicaciones de **múltiples contenedores.**
- Especificación en un archivo YAML.
- Comandos para administrar el ciclo completo de tu aplicación:
  - *Iniciar, detener y reconstruir servicios.*
  - *Ver estado de servicios en ejecución.*
  - *Transmisión de la salida de logs de servicios en ejecución.*
  - *Ejecución de comandos en un único servicio.*

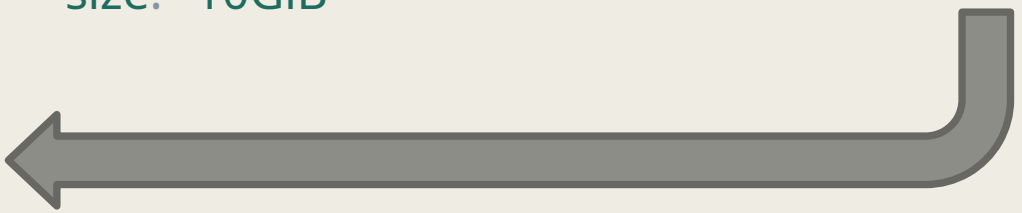
Docker Compose: casos de uso

- Entornos de desarrollo: Inicializa tu desarrollo con una instrucción.
  - *docker compose up*
- Entornos para pruebas automatizadas:
  - *docker compose up –d*
  - *./run_test*
  - *docker compose down*
- Implementaciones en producción de host único

Docker compose: Ejemplo

```
services:
    frontend:
        image: example/webapp
        ports: - "443:8043"
        networks:
            - front-tier
            – back-tier
    backend:
        image: example/database
        volumes:
            - db-data:/etc/data
        networks: - back-tier
volumes:
    db-data:
        driver: flocker
        driver_opts:
            size: "10GiB"
```

```
networks:
    front-tier: {}
    back-tier: {}
```



Docker Compose: Overview

- https://docs.docker.com/compose/reference/
- Sample apps: https://github.com/docker/awesome-compose