

Estrategias de Continuous Deployment

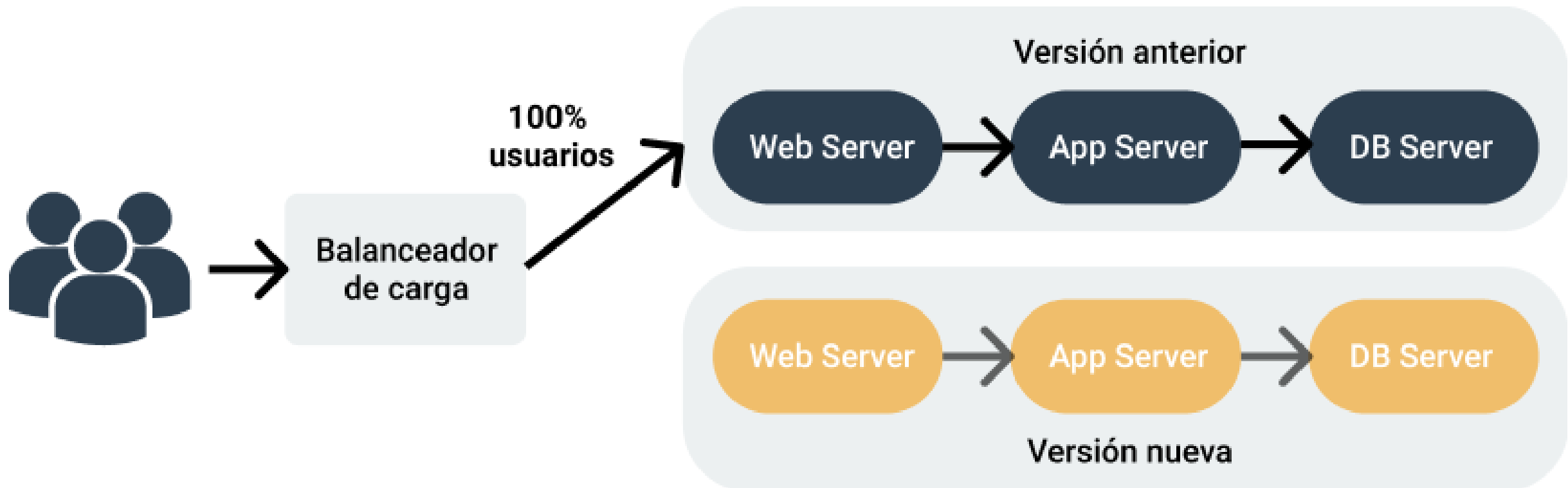
¿POR QUÉ USARLAS?

El despliegue en producción puede ser arriesgado. A pesar de todas las estrategias de mitigación que usamos (especialistas en control de calidad, set de tests automatizados, monitoreo y alertas, revisiones de código y análisis estático), los sistemas son cada vez más complejos. Esto hace que sea más probable que una nueva funcionalidad impacte provocando errores imprevistos y se deteriore la confianza de los clientes. Existen muchas estrategias que se pueden emplear, a continuación se describen dos de las más utilizados.

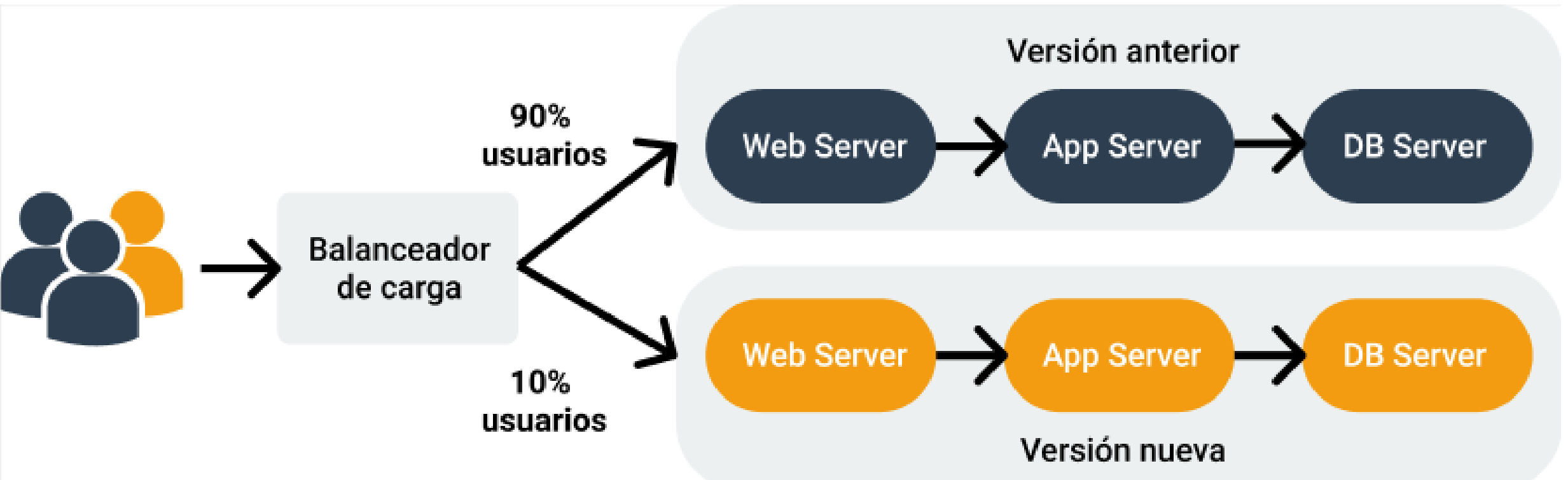
Canary Deployment

'Canary' por la estrategia que se utilizaba en la minería, donde llevaban canarios a las minas de carbón: si había un alto nivel de gases tóxicos el canario moriría, alertando así a los mineros. De forma similar, se despliega a un pequeño grupo de usuarios, si algo sale mal se vuelve a la versión anterior.

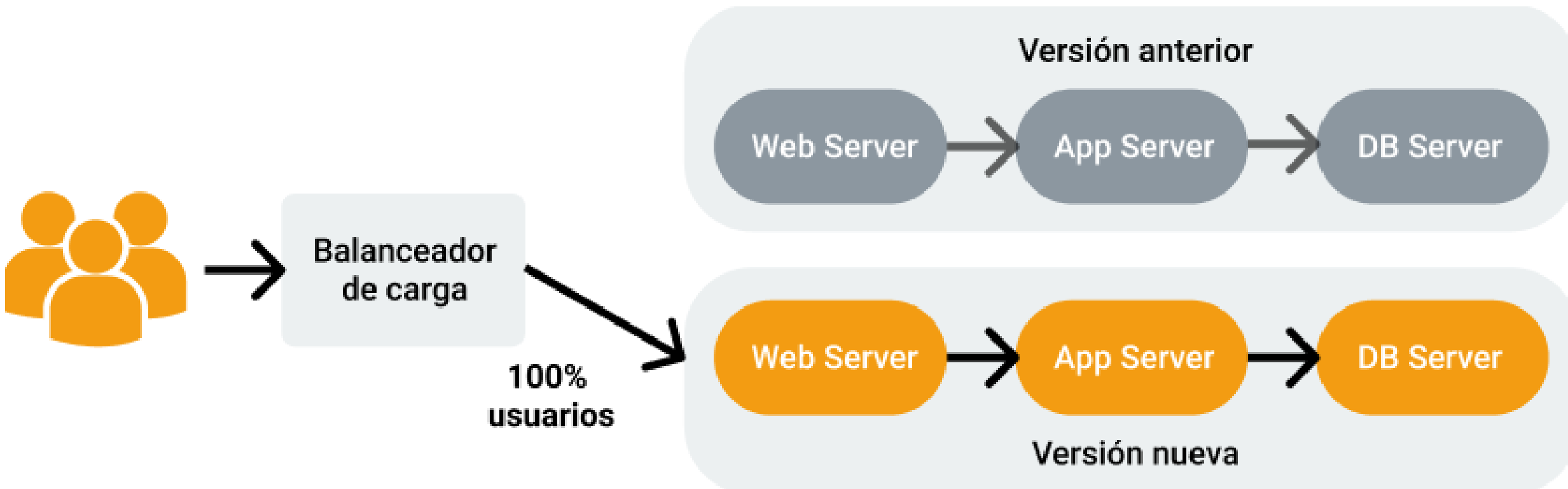
Se crean dos entornos de producción iguales. Se inicia enviando todo el tráfico a uno de ellos y agregando la nueva funcionalidad en el otro.



Se envía para un pequeño porcentaje de usuarios la nueva versión manteniendo el resto en la versión anterior.



Si ese grupo de usuarios no informa ningún error, se despliega gradualmente a más usuarios, hasta que todos usen la nueva versión.



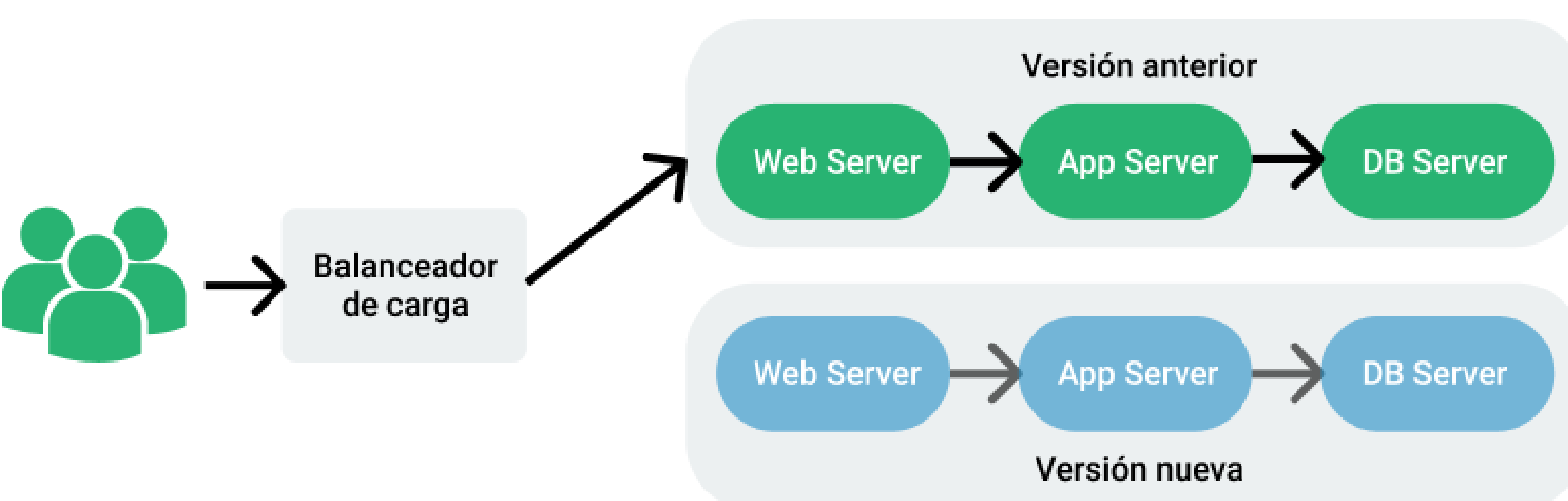
Útil cuando el equipo no está tan seguro de la nueva versión y no les importa un despliegue más lento si eso significa que podrán detectar los errores. Es actualmente la forma más utilizada para desplegar en producción.

- Bajo riesgo en relación con otras estrategias.
- No posee tiempos de inactividad.
- Pruebas en producción con usuarios reales.
- Económico, no es necesario mantener dos entornos de producción.
- Rollback rápido y seguro.
- La creación de scripts en la implementación puede ser compleja.
- Se necesita de herramientas e instrumentos para monitorear las pruebas en producción.

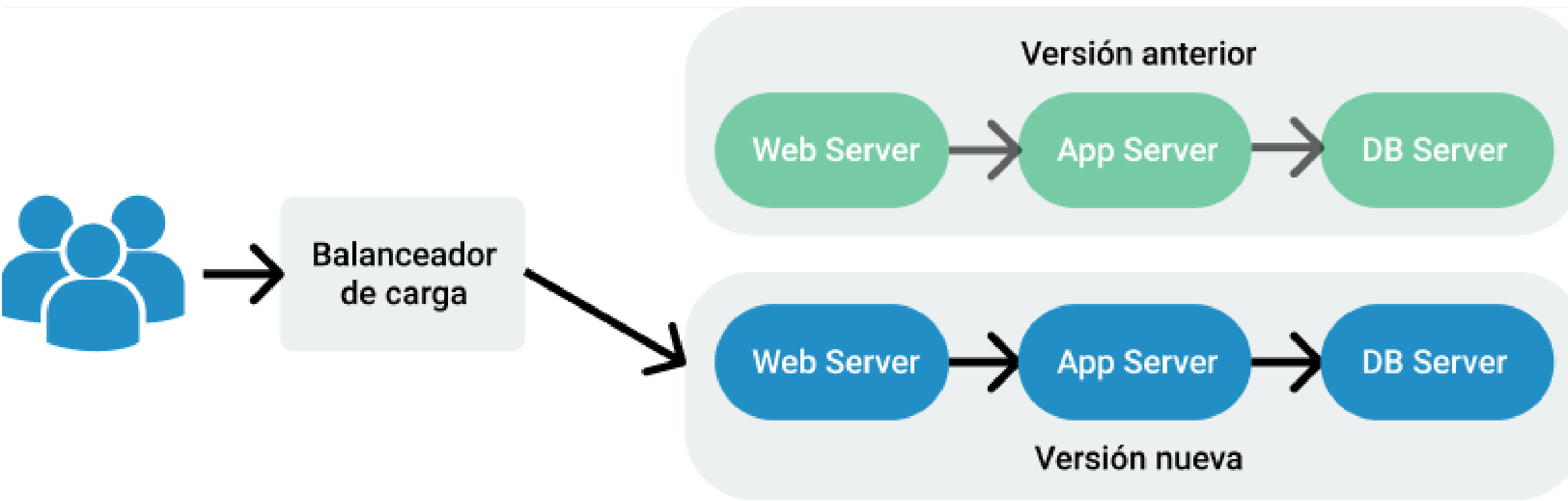
Blue Green Deployment

'Blue Green' porque la idea era evitar usar nombres que dieran a entender que un entorno era 'mejor' que el otro: usando colores no habrá un entorno que sea el 'mejor'.

Dos ambientes idénticos llamados "blue" (también conocido como staging) y "green" (también conocido como producción) se ejecutan simultáneamente con diferentes versiones o servicios / artefactos. QA y UAT se realizan normalmente en el ambiente azul.



Cuando está satisfecho, el tráfico se invierte (a través de un balanceador de carga) del ambiente verde (versión actual) al azul (nueva versión). Luego, puede retirar el ambiente anterior una vez que el despliegue se haya realizado correctamente.



Útil cuando es seguro que la nueva versión funcionará correctamente y desean una estrategia simple y rápida para implementarla.

- Rápido y fácil de implementar.
- Bajo riesgo en relación con otras estrategias.
- Retroceso rápido (se vuelve el tráfico al entorno anterior).
- Replicar un entorno de producción puede ser costoso.
- Una interrupción podría tener un impacto comercial a gran escala antes de que comience la reversión.
- Posible pérdida de transacciones de usuario cuando se realiza el cambio.
- Problemas de compatibilidad de la base de datos.

ORIGEN

CARACTERÍSTICAS

USOS



¿QUÉ ESTRATEGIA DEBO UTILIZAR?

Depende del tipo de aplicación/servicio y del entorno. La mayoría de los equipos utilizan actualmente Blue Green o Canary para aplicaciones web de naturaleza crítica. En muchos casos, los clientes mitigan de Blue Green a Canary para poder realizar pruebas en producción con un impacto comercial mínimo. También se pueden combinar las estrategias de despliegue en una sola.

LINKS

Harness. Canary Release vs Blue Green <https://harness.io/2018/02/blue-green-vs-canary-deployments/>
Rollout. Canary Deployment <https://rollout.io/blog/canary-deployment/>
Split. Canary Deployment <https://www.split.io/glossary/canary-deployment> - BlueGreen Deployment <https://www.split.io/glossary/blue-green-deployment>
Martin Fowler. Canary Deployment <https://martinfowler.com/bliki/CanaryRelease.html> - BlueGreen Deployment <https://martinfowler.com/bliki/BlueGreenDeployment.html>

GRUPO 3
Arancibia, Ana Luz (60279)
Becher, Priscila (61435)
Lizarraga, Jesús (62121)
Nuñez, Agustina (62809)
Palacios, Agustín (67960)

INGENIERÍA DE
SOFTWARE
4 K 3

UTN FACULTAD REGIONAL CORDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN