

Estrategias de implantación

Traducido automáticamente del Inglés

Visión general

Una estrategia de despliegue de Kubernetes define el ciclo de vida de una aplicación que alcanza y mantiene el estado configurado para objetos y aplicaciones de forma automatizada. Las estrategias de despliegue eficaces minimizan el riesgo.

Las estrategias de despliegue de Kubernetes se utilizan para

- Desplegar, actualizar o revertir ReplicaSets, Pods, Servicios y Aplicaciones.
- Pausar/reanudar despliegues
- Escalar despliegues manual o automáticamente

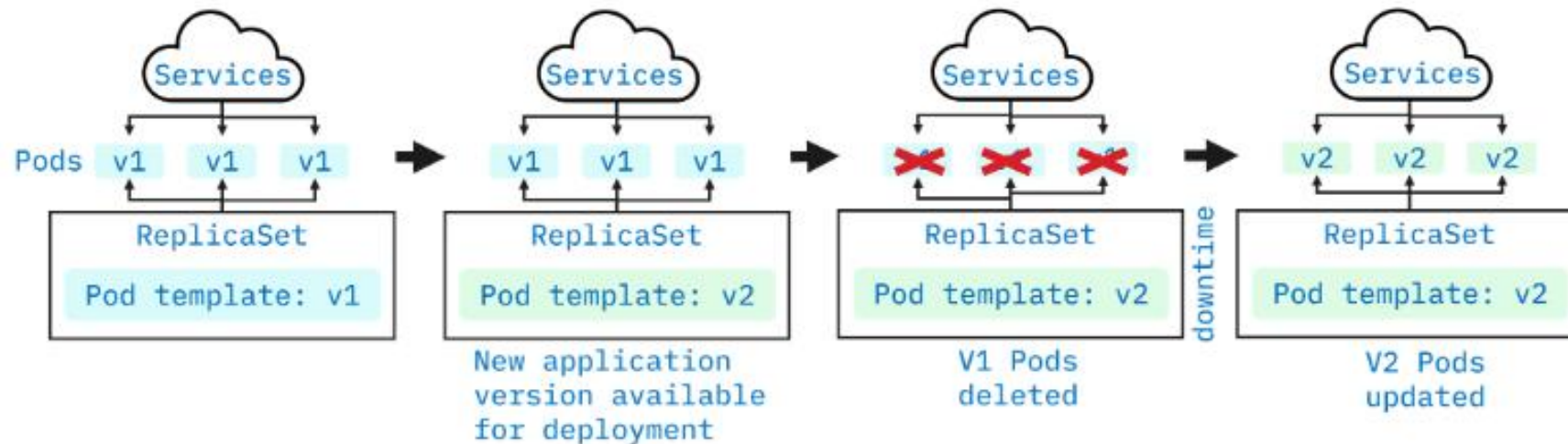
Tipos de estrategias de despliegue

Los siguientes son seis tipos de estrategias de despliegue:

1. Recrear
2. Enrollable
3. Azul/verde
4. Canario
5. Pruebas A/B
6. Sombra

Puede utilizar una única estrategia de despliegue o una combinación de varias estrategias de despliegue.

Estrategia de recreación



En la estrategia Recreate, los Pods que ejecutan la versión en vivo de la aplicación se apagan simultáneamente y se despliega una nueva versión de la aplicación en Pods recién creados.

Recreate es la estrategia de despliegue más sencilla. Hay un breve periodo de inactividad entre el cierre del despliegue existente y el nuevo despliegue

Los pasos de la estrategia Recreate incluyen:

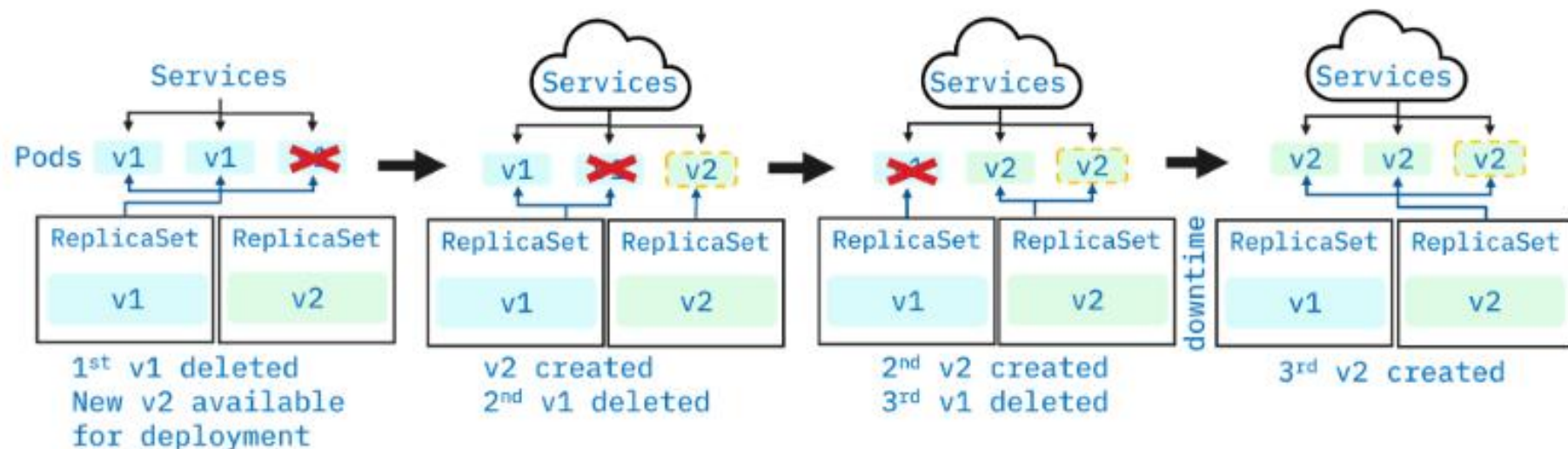
1. Una nueva versión de la aplicación (v2) está lista para su despliegue.
2. Todos los Pods que ejecutan la versión actual (v1) se apagan o eliminan.
3. Se crean nuevos Pods (v2).

El proceso de rollback se completa en orden inverso, reemplazando la versión 2 (v2) por la versión 1 (v1).

El proceso de rollback se completa en orden inverso, reemplazando la versión 2 (v2) por la versión 1 (v1).

Ventajas	Contras
Configuración sencilla	Se produce un breve tiempo de inactividad entre el apagado y el nuevo despliegue
Sustitución completa de la versión de la aplicación	

Estrategia escalonada



En una estrategia progresiva, cada Pod se actualiza de uno en uno. Un único Pod v1 se sustituye por un nuevo Pod v2. Cada Pod v1 se actualiza de este modo hasta que todos los Pods son v2. Durante una actualización de estrategia rodante, apenas hay tiempo de inactividad, ya que los usuarios son dirigidos a cualquiera de las dos versiones.

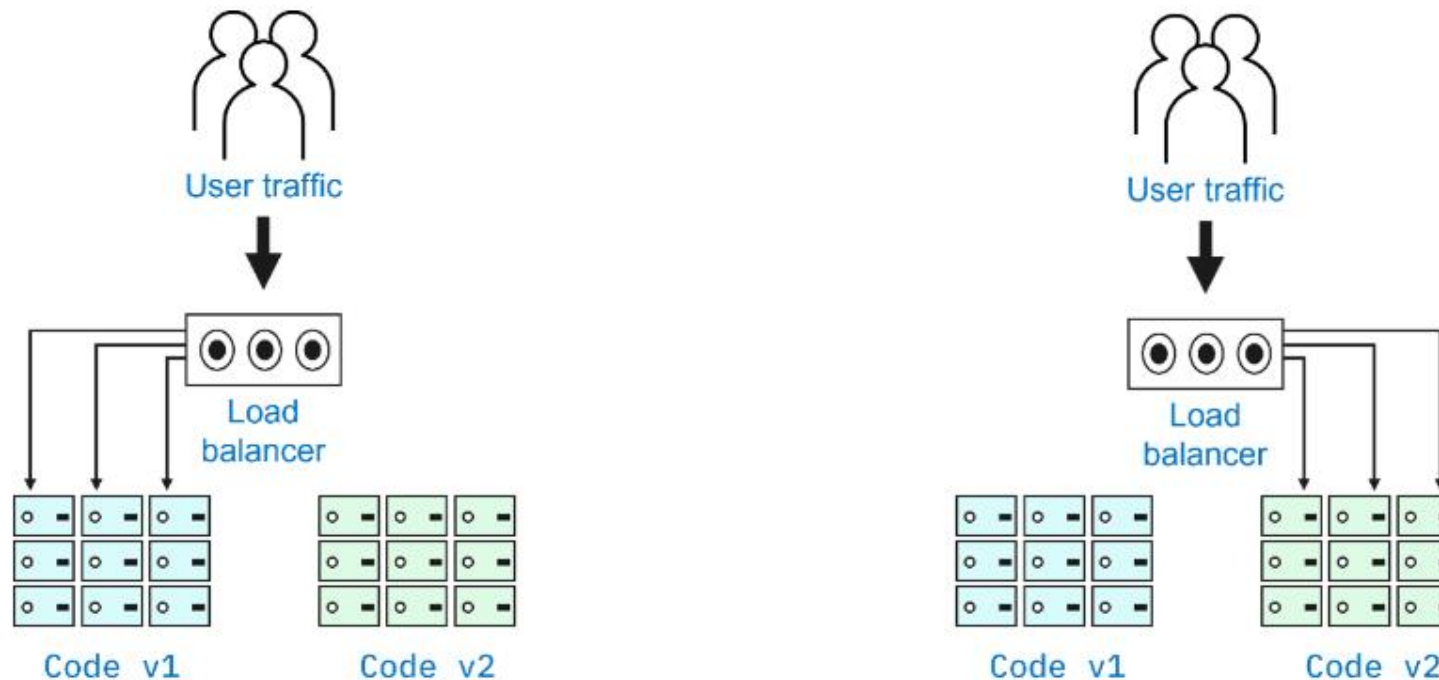
Los pasos de la estrategia rodante incluyen:

1. Una nueva versión de la aplicación (v2) está lista para su despliegue.
2. Uno de los Pods que ejecuta la versión actual (v1) se apaga o se elimina.
3. Se crea un nuevo Pod (v2) para sustituir al Pod (v1) que se eliminó.
4. Los pasos 2 y 3 se repiten hasta que todos los Pods (v1) son eliminados y reemplazados por Pods (v2).

El proceso de reversión se invierte, y los Pods v2 son reemplazados por Pods v1.

Pros	Contras
Configuración sencilla	El rollout/rollback lleva tiempo
Adecuado para aplicaciones con estado que necesitan manejar el reequilibrio de los datos	No se puede controlar la distribución del tráfico

Estrategia azul/verde



En una estrategia azul/verde, el entorno azul es la versión activa de la aplicación. El entorno verde es una copia exacta que contiene el despliegue de la nueva versión de la aplicación. El entorno verde se prueba a fondo. Una vez resueltos todos los cambios, errores y problemas, el tráfico de usuarios pasa del entorno azul al verde.

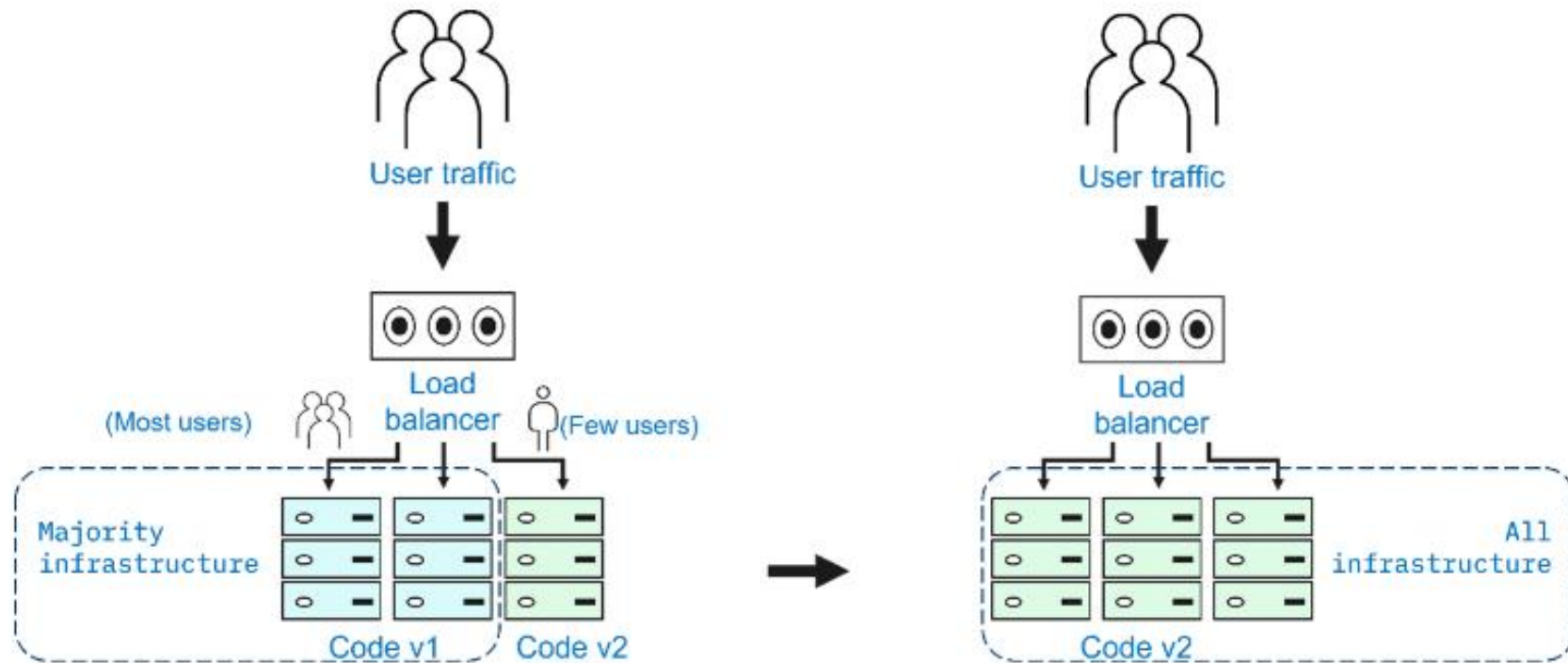
Los pasos de la estrategia azul/verde incluyen:

1. Crear un nuevo entorno idéntico al entorno de producción actual.
2. Diseñar la nueva versión y probarla a fondo hasta que esté lista para la producción.
3. Dirigir todo el tráfico de usuarios a la nueva versión.

Para realizar una reversión, vuelva a cambiar los entornos.

Ventajas	Contras
Despliegue/repliegue instantáneo (sin tiempo de inactividad)	Caro (requiere el doble de recursos)
La nueva versión está disponible inmediatamente para todos los usuarios	Se requieren pruebas rigurosas antes de pasar a producción
	El manejo de aplicaciones con estado es difícil

Estrategia canaria



En una estrategia canaria, la nueva versión de la aplicación se prueba utilizando un pequeño grupo de usuarios aleatorios junto con la versión actual de la aplicación. Una vez que la nueva versión de la aplicación se ha probado con éxito, se distribuye a todos los usuarios.

Los pasos de la estrategia canaria incluyen:

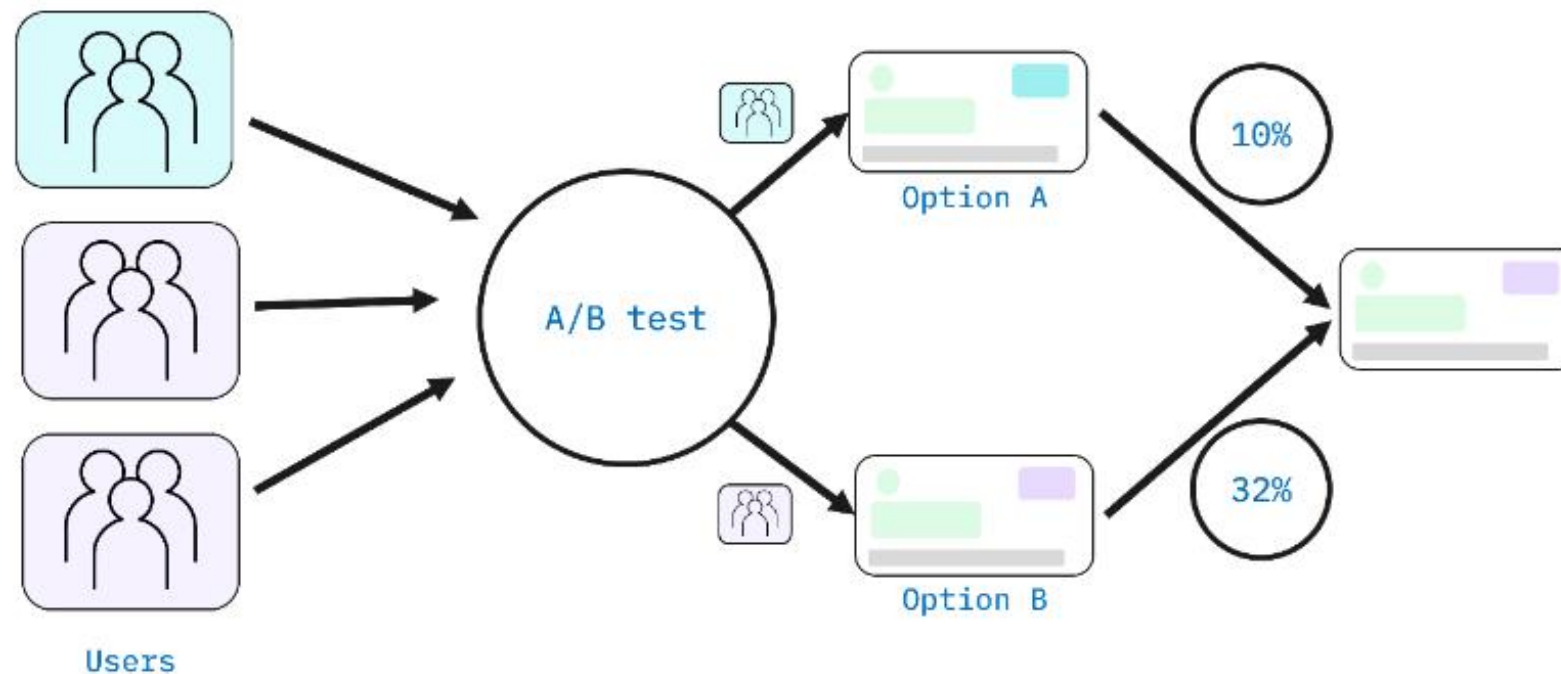
1. Diseñar una nueva versión de la aplicación.
2. Encaminar una pequeña muestra de solicitudes de usuarios a la nueva versión.
3. Pruebe la eficacia, el rendimiento, los errores y los problemas, y revierta la aplicación según sea necesario.
4. Repita los pasos 1 a 3. Una vez resueltos todos los problemas, dirija todo el tráfico a la nueva versión.

La reversión no requiere tiempo de inactividad, ya que pocos usuarios están expuestos a la nueva versión.

La reversión no requiere tiempo de inactividad, ya que pocos usuarios están expuestos a la nueva versión.

Ventajas	Contras
Conveniente para controlar la fiabilidad, los errores y el rendimiento	Despliegue lento, acceso gradual de los usuarios
Reversión rápida	

Estrategia de Pruebas A/B



La estrategia de Pruebas A/B, también conocida como Pruebas divididas, evalúa dos versiones de una aplicación (versión A y versión B). Con las Pruebas A/B, cada versión tiene características que se adaptan a diferentes grupos de usuarios. En función de la interacción y los comentarios de los usuarios, se puede seleccionar la mejor versión para su despliegue global

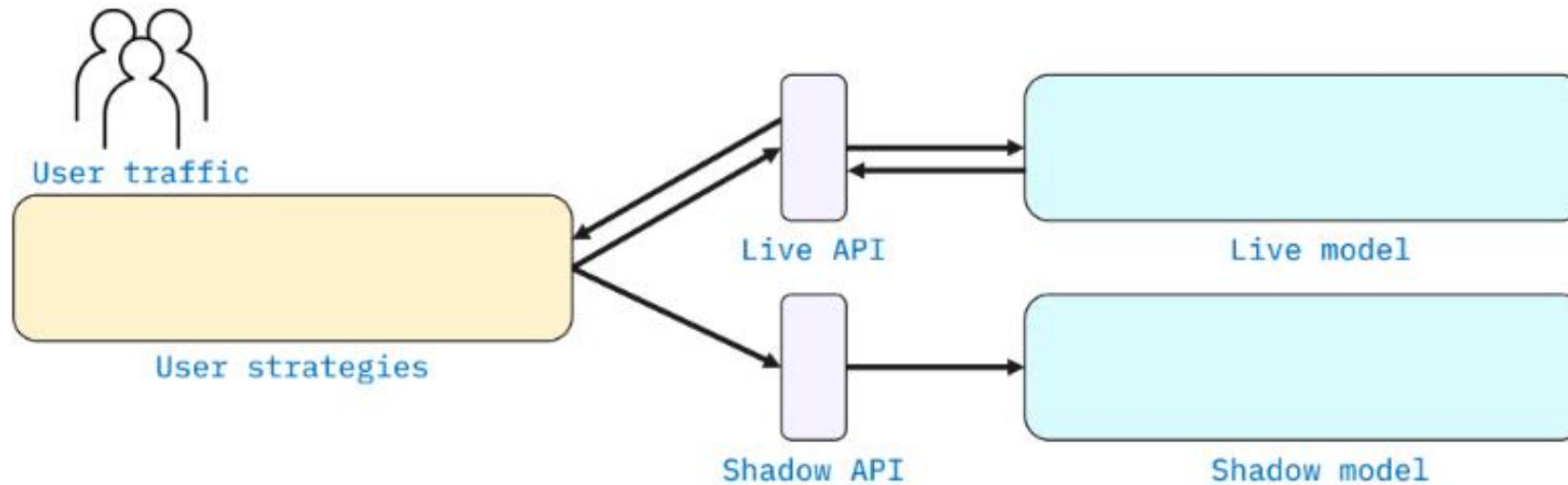
Los pasos de la estrategia de Pruebas A/B incluyen:

1. Diseñar una nueva versión de la aplicación añadiendo sobre todo funciones de IU.
2. Identificar un pequeño conjunto de usuarios en función de condiciones como el peso, el valor de la cookie, los parámetros de consulta, la geolocalización, la versión del navegador, el tamaño de la pantalla, el sistema operativo y el idioma.
3. Dirigir las peticiones del conjunto de usuarios a la nueva versión.
4. Compruebe si hay errores, eficacia, rendimiento y problemas.
5. Una vez resueltos todos los problemas, dirija todo el tráfico a la nueva versión.

Se pueden aplicar retrocesos, pero el tiempo de inactividad puede afectar al usuario.

Ventajas	Contras
Pueden ejecutarse varias versiones en paralelo	Requiere un balanceador de cargas inteligente
Control total sobre la distribución del tráfico	Dificultad para solucionar los errores de una sesión determinada; el rastreo distribuido es obligatorio

Estrategia en la sombra



En una estrategia en la sombra, se despliega una "versión en la sombra" de la aplicación junto con la versión activa. Las peticiones de los usuarios se envían a ambas versiones, y ambas gestionan todas las peticiones, pero la versión en la sombra no devuelve las respuestas a los usuarios. Esto permite a los desarrolladores ver cómo funciona la versión en la sombra utilizando datos del mundo real sin interrumpir la experiencia del usuario.

Para realizar una reversión, vuelve a cambiar los entornos.

Ventajas	Contras
Pruebas de rendimiento con tráfico de producción	Costoso (duplicación de recursos)
Sin impacto en el usuario	No es una verdadera prueba de usuario, puede dar lugar a resultados malinterpretados
Sin tiempo de inactividad	Configuración compleja
	Requiere la supervisión de dos entornos