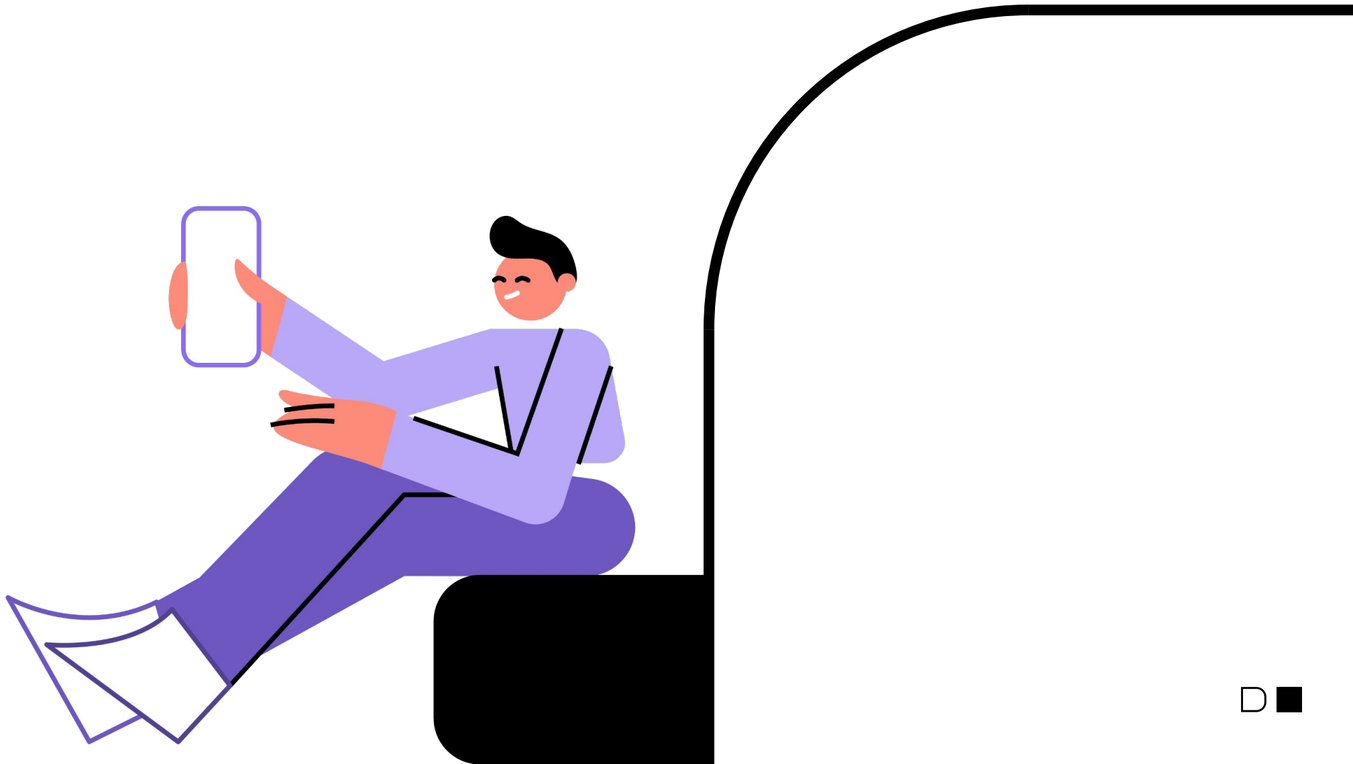
DIGITALHOUSE

Escalado

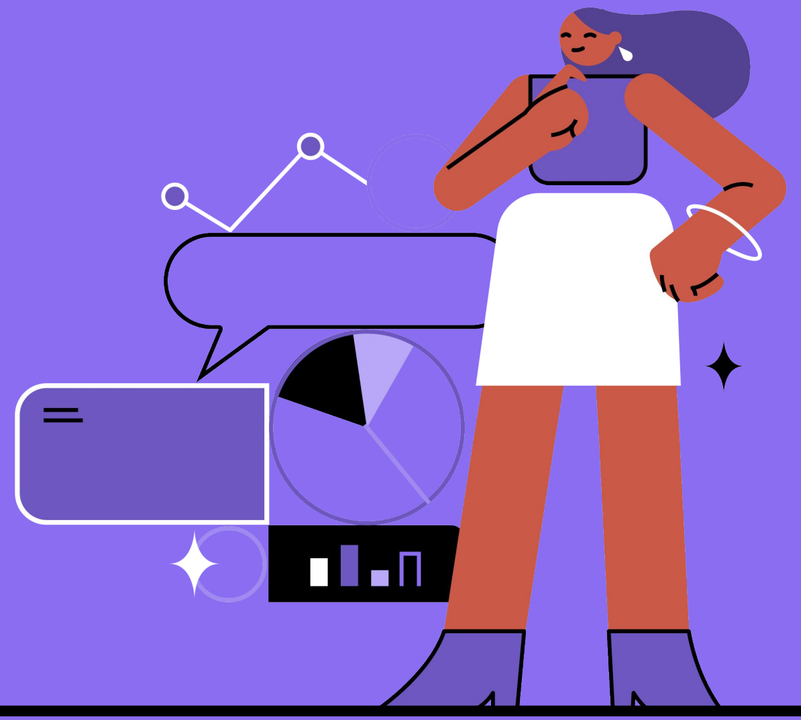
Certified Tech Developer
The Ultimate Degree

Índice

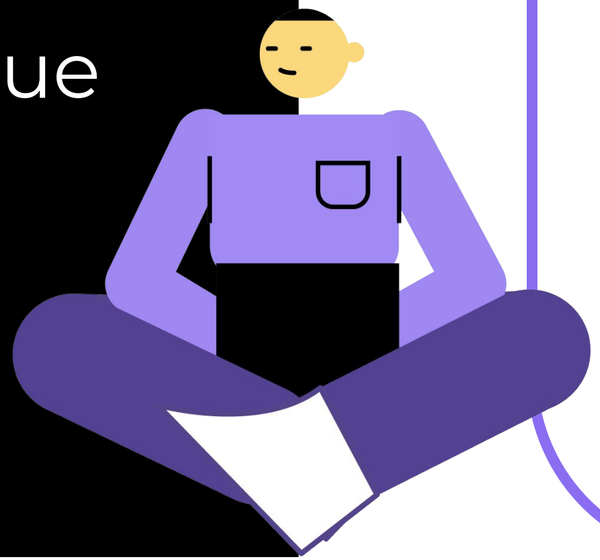


01 ¿Qué es la escalabilidad?

La escalabilidad es un componente esencial del software empresarial. Priorizarlo desde el principio conduce a menores costos de mantenimiento, una mejor experiencia del usuario y una mayor agilidad.



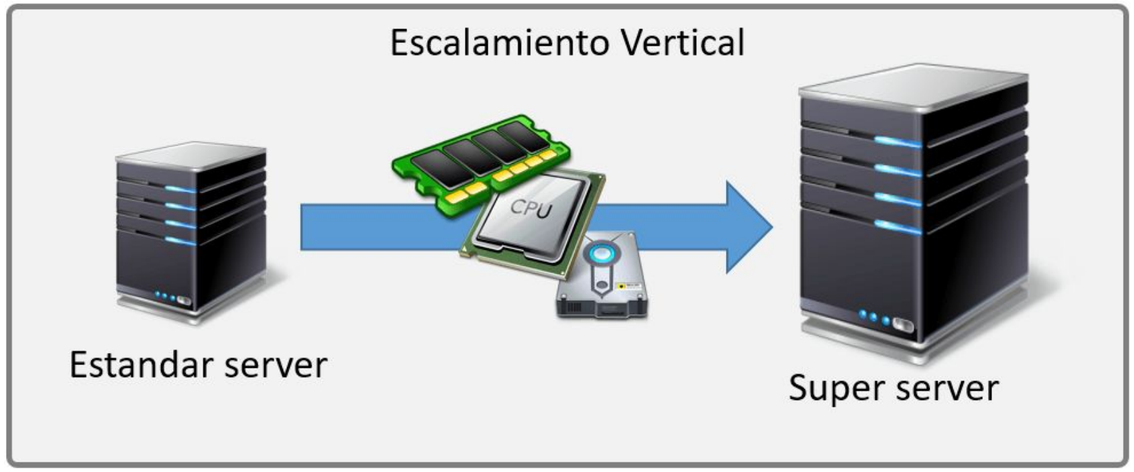
Es la capacidad de adaptación y respuesta de un sistema con respecto al rendimiento del mismo a medida que aumentan de forma significativa el número de usuarios del mismo.



02 Tipos de escalados

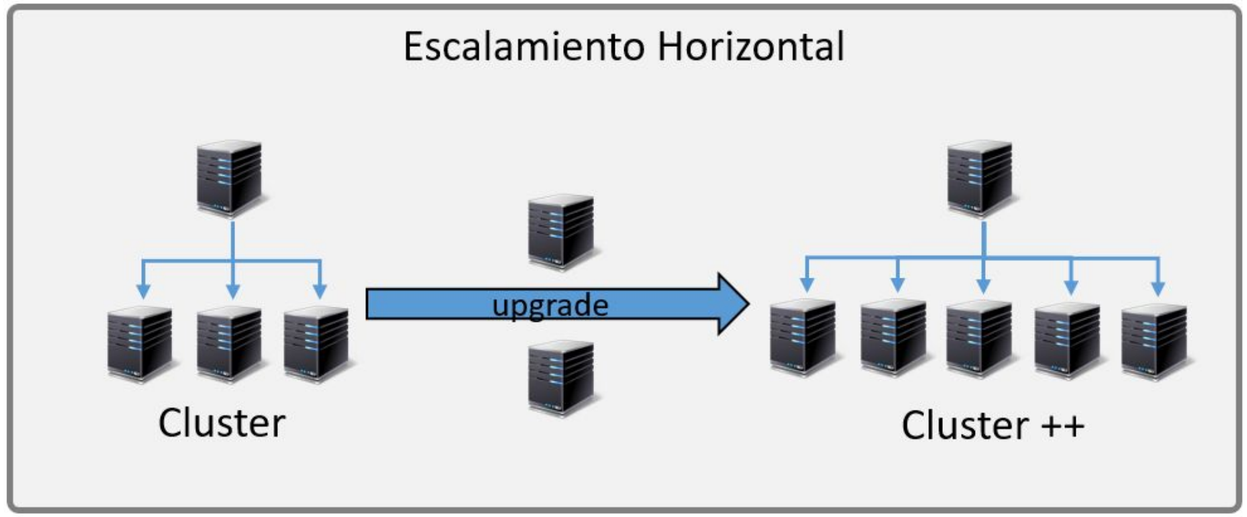
Escalabilidad vertical

Significa hacer crecer el hardware, es decir, aumentar sus componentes como disco duro, memoria, procesador, etc. Pero también puede ser la migración completa del hardware por uno más potente. El esfuerzo de este crecimiento es mínimo, pues no tiene repercusiones en el software, ya que solo será respaldar y migrar los sistemas al nuevo hardware.



Escalabilidad horizontal

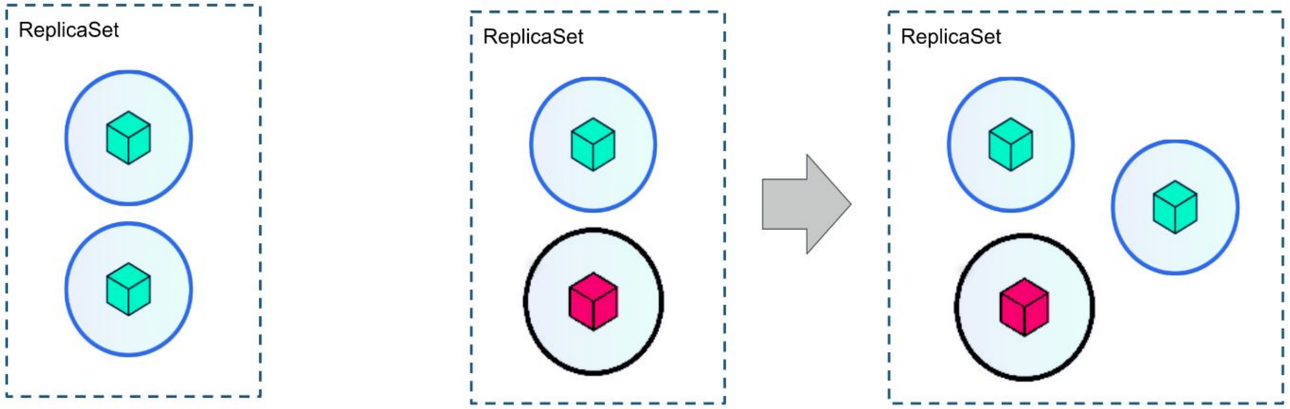
Es sin duda el más potente, pero también el más complicado. Este modelo implica tener varios servidores trabajando como un todo. Se crea una red de servidores, con la finalidad de repartirse el trabajo. Cuando la performance se ve afectada con el incremento de usuarios, se añaden nuevos servidores para repartir la carga de trabajo.



03 ReplicaSet

¿Qué es?

Es un objeto de kubernetes que permite escalar horizontalmente un pod o grupo de pods. Se define generalmente en un archivo YAML y contiene campos específicos como, por ejemplo, el campo **selector** que indica cómo identificar a los pods que puede adquirir. Alcanza así su propósito mediante la creación y eliminación de los pods que sean necesarios para alcanzar el número esperado. Se puede aumentar o reducir fácilmente un ReplicaSet, simplemente actualizando el campo .spec.replicas. El controlador del ReplicaSet se asegura de que el número deseado de pods con un selector de etiquetas coincidente está disponible y operacional.



04 Deployment

¿ReplicaSet o Deployment?

Digamos que venimos usando el ReplicaSet-A para controlar un grupo de pods y necesitamos actualizarlos a una versión más nueva. Entonces deberíamos crear el Replicaset-B, reducir el ReplicaSet-A y aumentar ReplicaSet-B (este proceso se conoce como Rolling update). Aunque esto hace el trabajo, no es una buena práctica y es mejor dejar que K8S haga el trabajo.

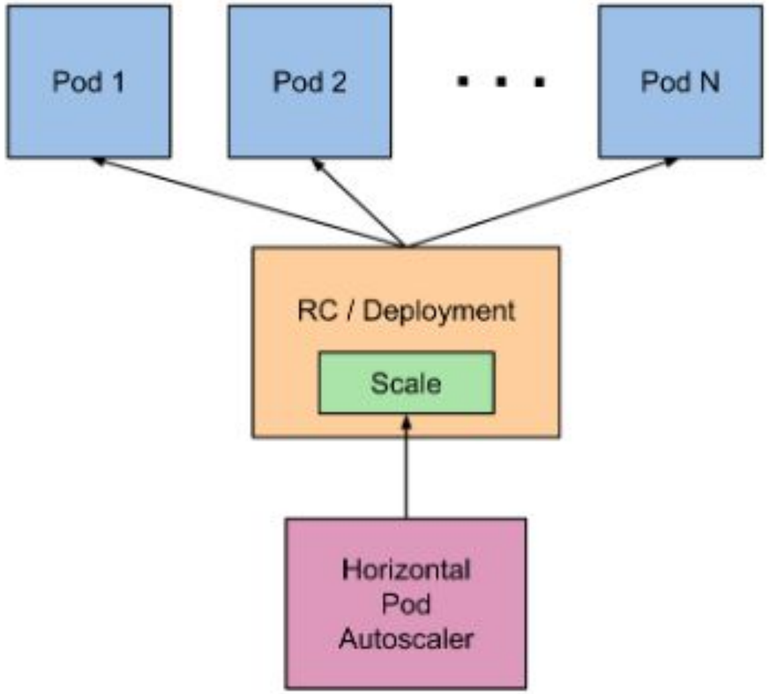
Para ello existe el objeto Deployment, se implementa generalmente con un archivo YAML y hace el Rolling update sin interacción humana, agregando un nivel de abstracción al ReplicaSet, lo que simplifica la tarea de actualizar pods. Es una buena práctica utilizar Deployments para manejar los ReplicaSets.

05

HPA (horizontal pod autoscaler)

¿Qué es?

Un Horizontal Pod Autoscaler actualiza automáticamente un recurso de carga de trabajo (como un Deployment o ReplicaSet), con el objetivo de escalar automáticamente la carga para que coincida con la demanda.



Escalar en base a métricas

Cualquier objetivo de HPA se puede escalar en función del uso de recursos de los pods en el objetivo de escalado. Al definir la especificación del pod, se deben especificar las solicitudes de recursos como cpu y memoria. Esto se usa para determinar la utilización de recursos y lo usa el controlador HPA para escalar el objetivo hacia arriba o hacia abajo. Para usar el escalado basado en la utilización de recursos, hay que especificar una fuente de métrica como esta.

```
type: Resource
resource:
 name: cpu
 target:
 type: Utilization
 averageUtilization: 60
```

Con esta métrica, el controlador HPA mantendrá la utilización promedio de los pods en el objetivo de escalado en un 60 %.

DIGITALHOUSE

¡Muchas gracias!