Deployment Strategy

A screen shot of a computer

AI-generated content may be incorrect.

Say for example we have 5 replicas of our web application instance deployed.

When I say update, it could be different things such as updating our application version by updating the version of docker containers used, updating their labels or updating the number of replicas etc.

Once we make the necessary changes, we run the kubectl apply command to apply the changes. A new rollout is triggered, and a new revision of the deployment is created.

But there is another way to do the same thing. We could use the kubectl set image command to update the image of our application. But remember, doing it this way will result in the deployment-definition file having a different configuration. So, we must be careful when using the same definition file to make changes in the future.

**🧾 What is the Recreate strategy?**

In the **Recreate** strategy, **all existing Pods are first terminated** before **new Pods are created**.  
This approach ensures that **only one version of the application runs at a time**.

This contrasts with the default **RollingUpdate** strategy, which gradually replaces old Pods with new ones.

**📦 When to use Recreate strategy?**

Use **Recreate** when:

* Your app **does not support running multiple versions** at once (e.g., stateful apps, DBs).
* There are **conflicts or constraints** with two versions running simultaneously (e.g., port conflicts).

🛠️ How to configure it?  
A screen shot of a computer program

AI-generated content may be incorrect.



For example, update the image name with **writer-app:v2.0**

A screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

🔁 This will first terminate the existing Pods of writer-pod, and only then create new ones.

**📌 Important Notes:**

* Downtime
  + Since all Pods are terminated first, expect **temporary unavailability**.
* No Overlapping Pods
  + Ensures **only one version** is ever running.
* No Need for maxSurge or maxUnavailable
  + These are **not applicable** with Recreate.

**🔁 What is Rolling Update?**

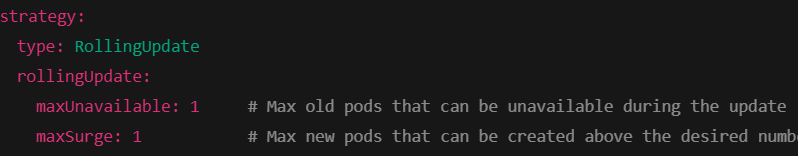
The **RollingUpdate** strategy is the **default update strategy** for Deployments in Kubernetes.

We do not destroy all old Pods at once. Instead, we take down the older version and bring up a newer version one by one (or batches). This way the application never goes down.

**🧠 How it works:**

1. Kubernetes **creates a new Pod** with the updated version.
2. Once the new Pod is **healthy**, it **terminates one old Pod**.
3. This process **repeats until all old Pods are replaced**.

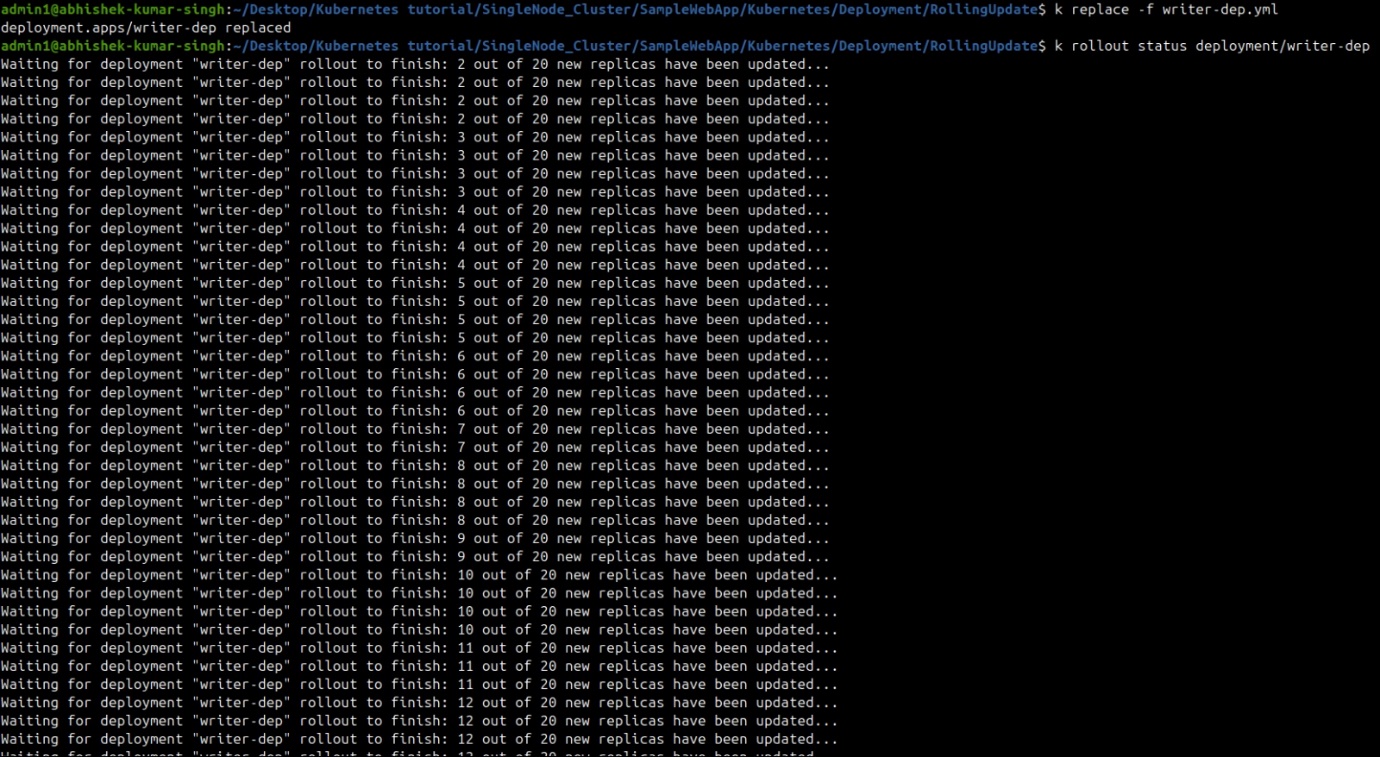
⚙️ Configuration (in YAML):



* These fields control **how many Pods** can be added or taken down **during the rollout**.
* We can use absolute numbers (1) or percentages (50%).

A black background with pink and green text

AI-generated content may be incorrect.



A screenshot of a computer

AI-generated content may be incorrect.

**📌 Important Notes:**

* **Zero downtime** deployment.
* Support Rollbacks:
  + Easy to **rollback** to the previous.  
    
* Health Checks Matter:
  + Rolling update **waits** for readiness probes to pass before proceeding.
  + If readiness fails, the rollout **pauses**, ensuring stability.
* Deployment History
  + 