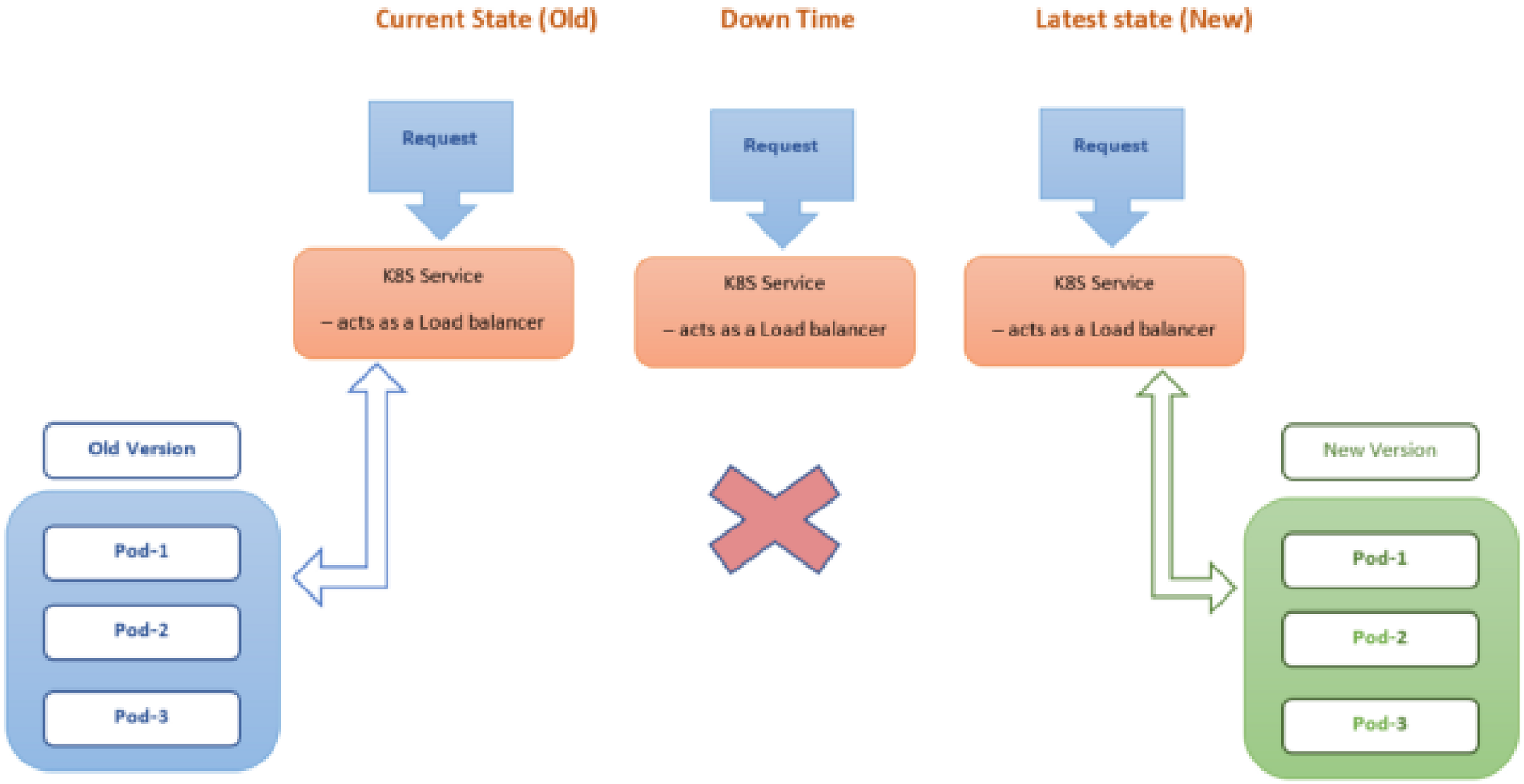
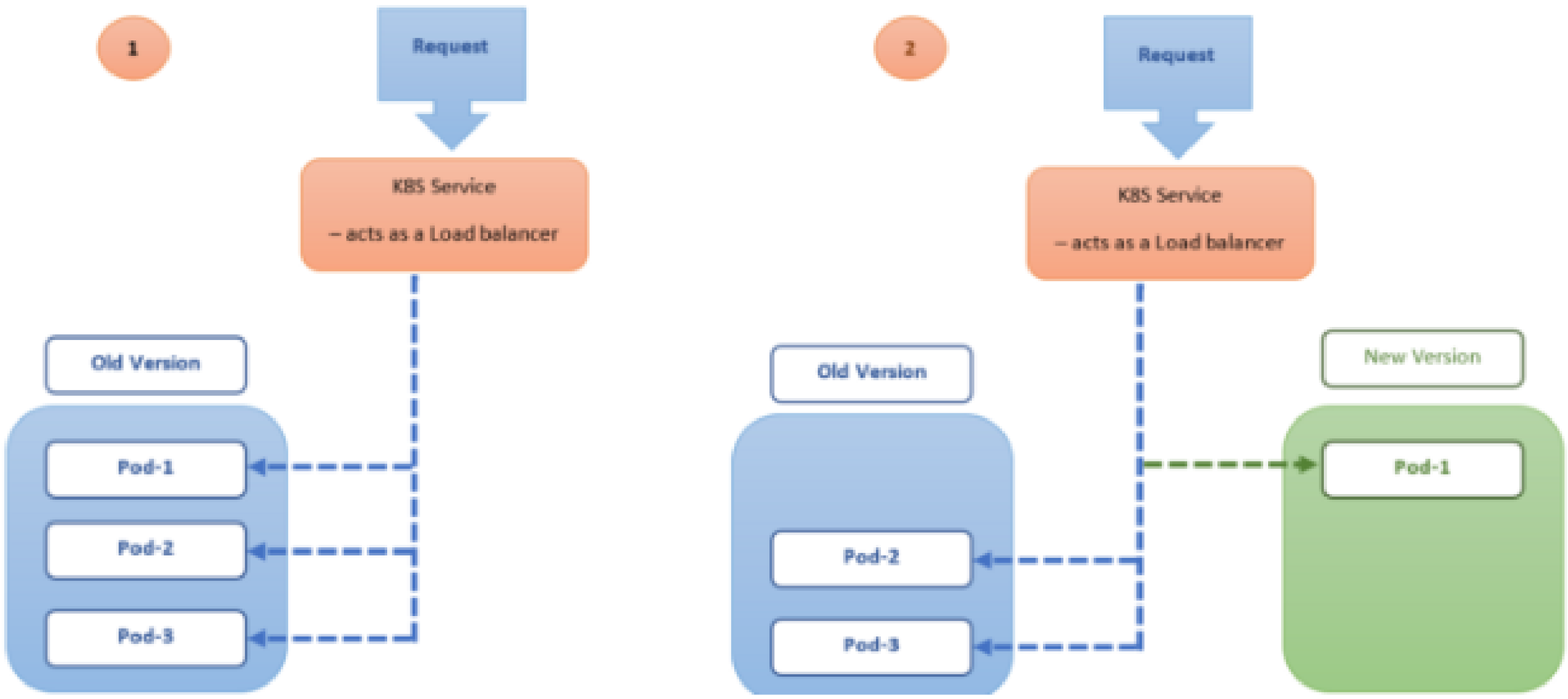
Kubernetes (K8S) Deployment Strategies

```
[...]
kind: Deployment
spec:
   replicas: 3
   strategy:
      type: Recreate
[...]
```

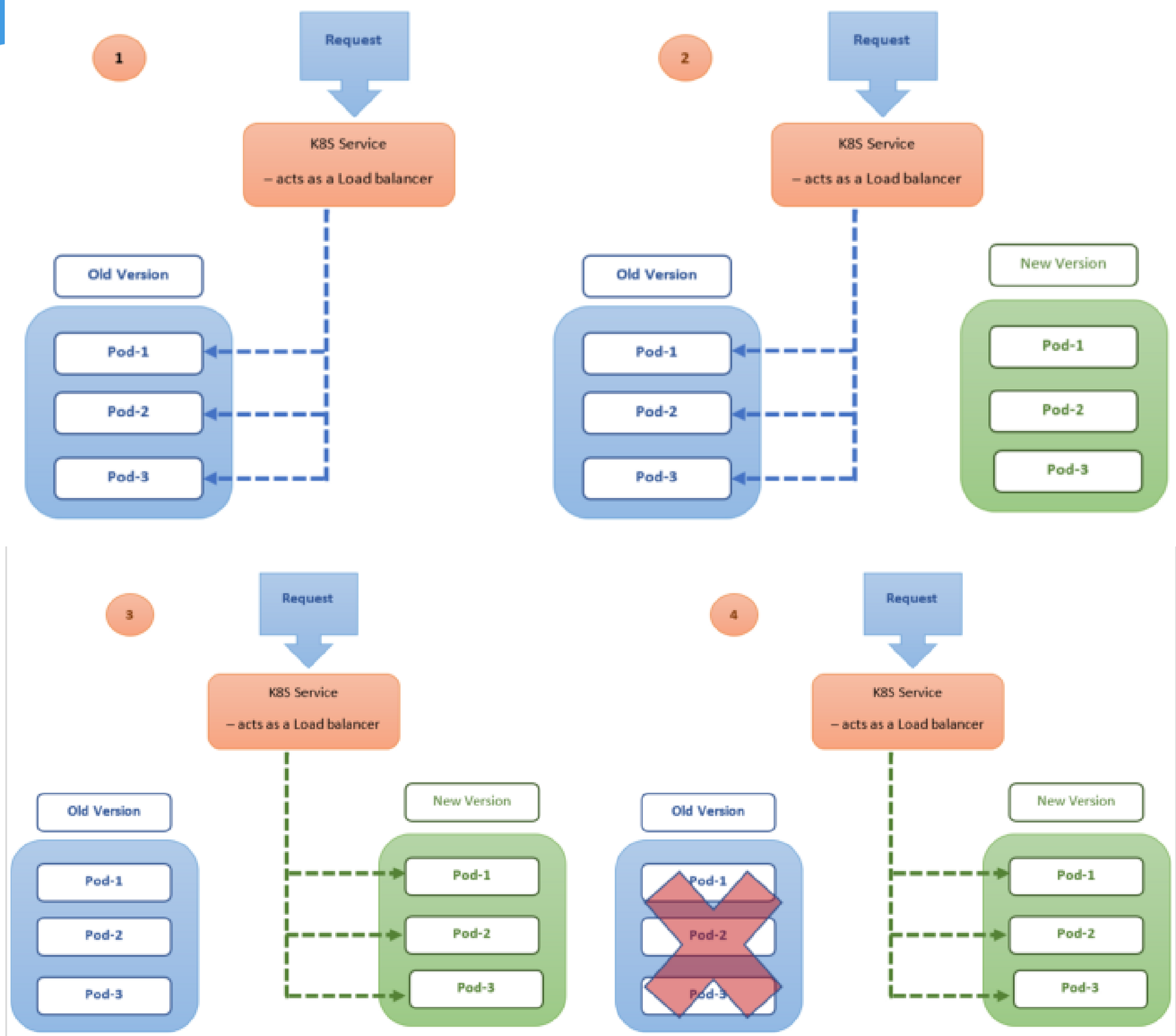


Recreate Deployment: In this deployment strategy, all of the old pods will be killed at once and replace all with new ones.

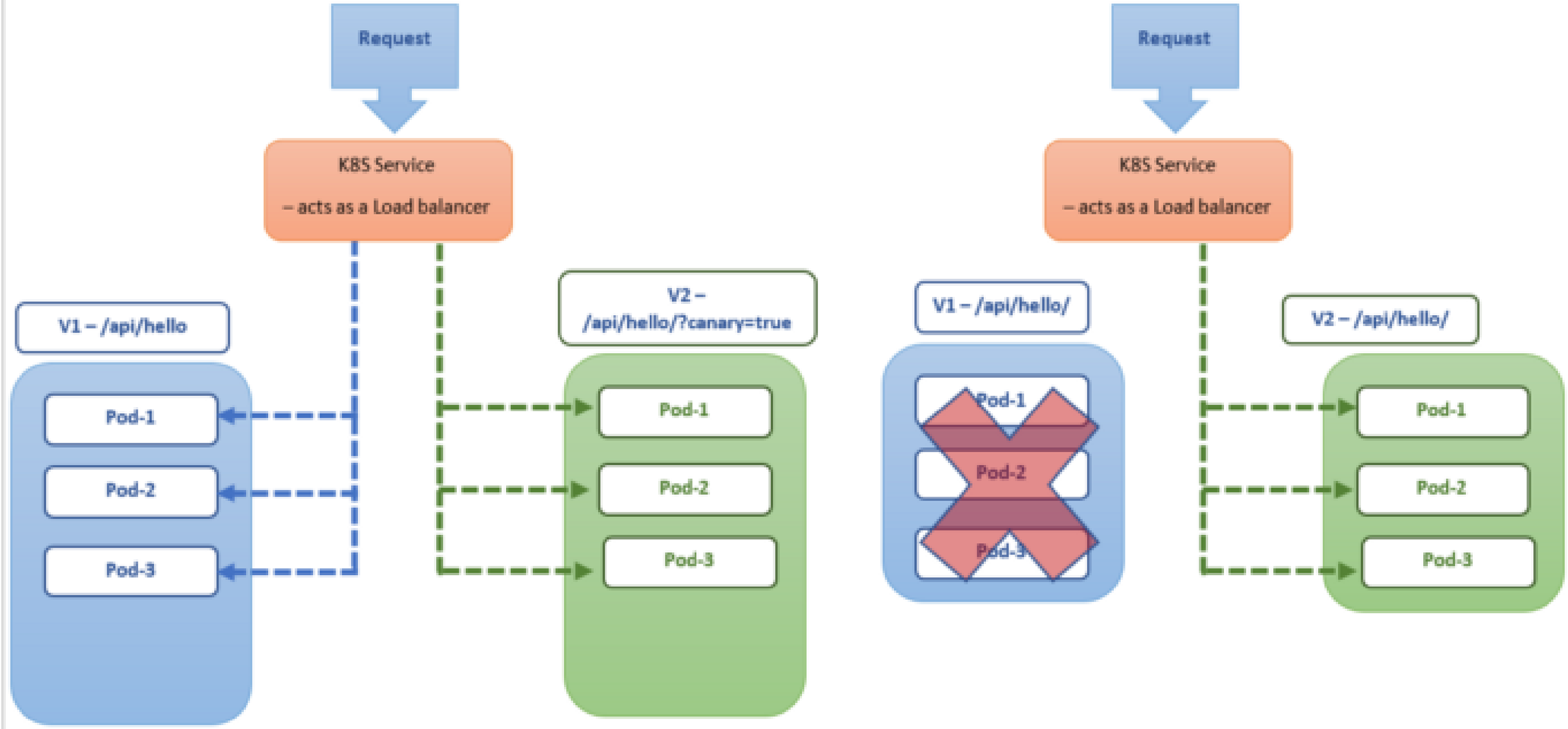
```
[...]
kind: Deployment
spec:
   replicas: 3
   strategy:
      type: RollingUpdate
[...]
```



Rolling update (or) Ramped slow-rollout Deployment: This strategy recreates the Pods in a rolling update or incremental fashion i.e., one after the other, here it will kill and recreate the pods without interrupting applications unavailability.



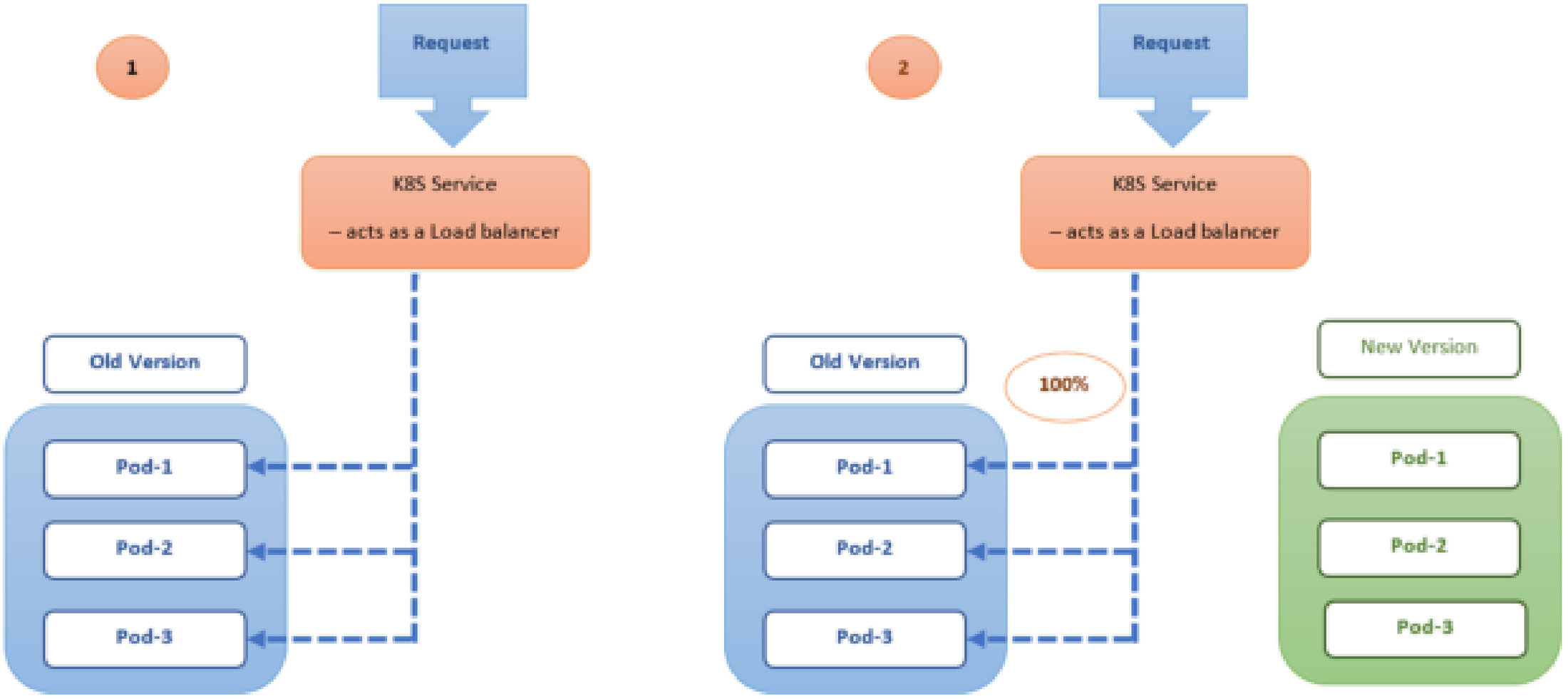
Blue/Green (or) Red/Black Deployment: Both old(blue) and new(green) application versions will be deployed alongside. Green deployments are available through a different service/port and it will be tested as per the requirements and replace blue with green deployments.



A/B Testing Deployment: Here, in A/B deployments we routes requests to a subset of users under specific conditions and can implemented by adding additional functionality on top Canary deployments.

```
Example:kind: RouteRule
metadata:
    name: shasr-app-v1
spec:
    destination:
        name: shasr-app
    route:
    - labels:
        version: v1.0.0
    match:
        request:
            headers:
                x-api-version:
                    exact: "v1.0.0"


kind: RouteRule
metadata:
    name: shasr-app-v2
spec:
    destination:
        name: shasr-app
    route:
    - labels:
        version: v2.0.0
    match:
        request:
            headers:
                x-api-version:
                    exact: "v2.0.0"
```



Canary Deployment: Canary deployments are very helpful for releasing and testing new functionality to a subset of users and and go with full rollout (when things works seamlessly with new version).

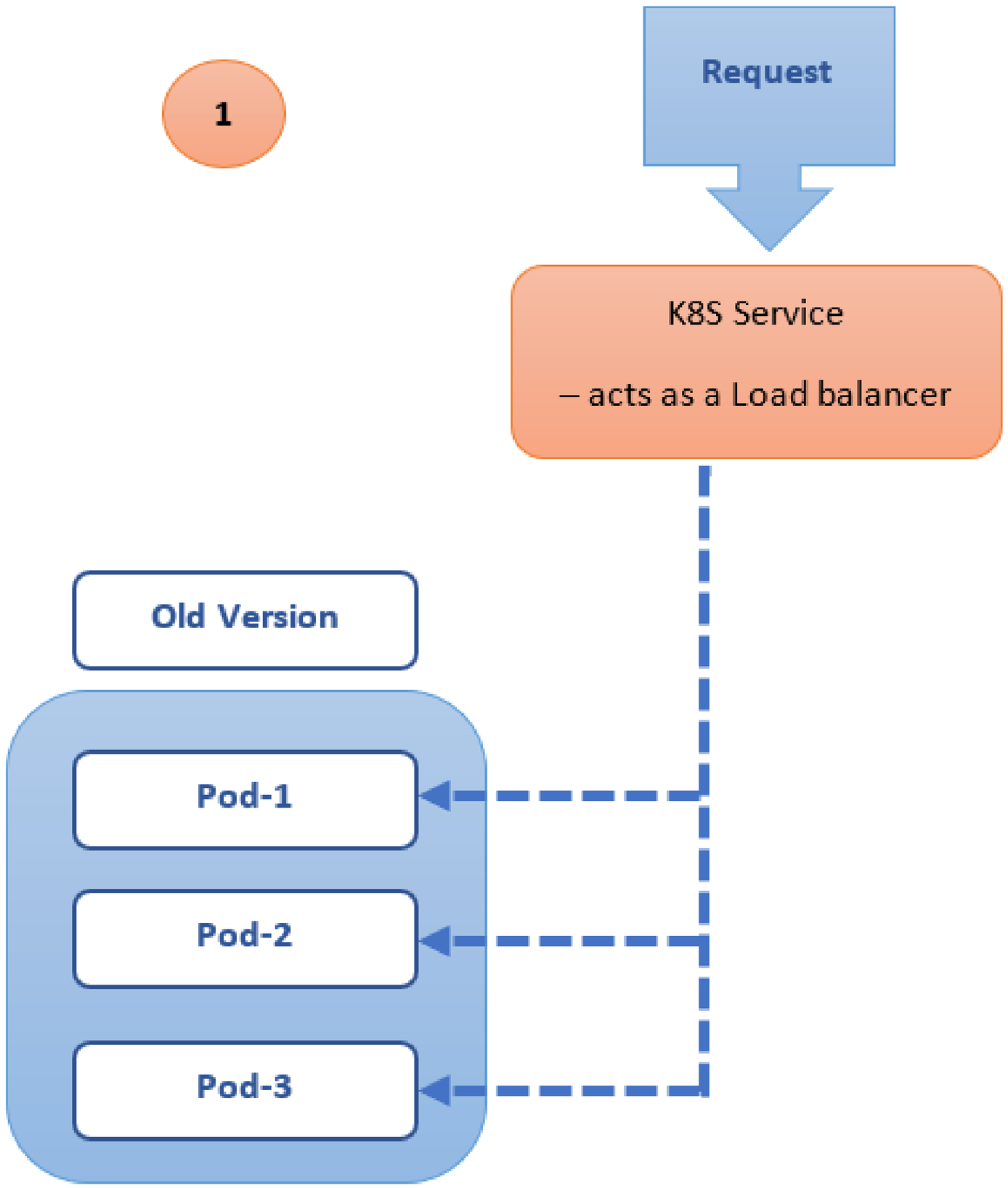
Example:

```
...
kind: Deployment
metadata:
    name: shasr-app-v1
spec:
    replicas: 5
    template:
        labels:
            app: shasr-app
            version: v1.0.0
...

...
kind: Deployment
metadata:
    name: shasr-app-v2
spec:
    replicas: 1
    template:
        labels:
            app: shasr-app
            version: v2.0.0
...

...
kind: Service
metadata:
    name: shasr-app
spec:
    selector:
        app: shasr-app
...

...
kind: RouteRule
metadata:
    name: shasr-app
    spec:
        destination:
            name: shasr-app
        route:
        - labels:
            version: v1.0.0
          weight: 95 # 95% traffic
        - labels:
            version: v2.0.0
          weight: 5 # 5% traffic
...
```



Shadow (or) Mirrored Deployment: Incoming traffic will be mirrored to a new version and doesn't impact the application traffic. It helps to test production traffic on a new version and can be rolled out based on the application performance/stability.

Example:

```
kind: RouteRule
spec:
    destination:
        name: shasr-app-v1
    route:
    - labels:
        version: v1.0.0
        weight: 100
    - labels:
        version: v2.0.0
        weight: 0
    mirror:
        name: shasr-app-v2
    labels:
        version: v2.0.0
```