



Introducción a las cargas de trabajo de Kubernetes



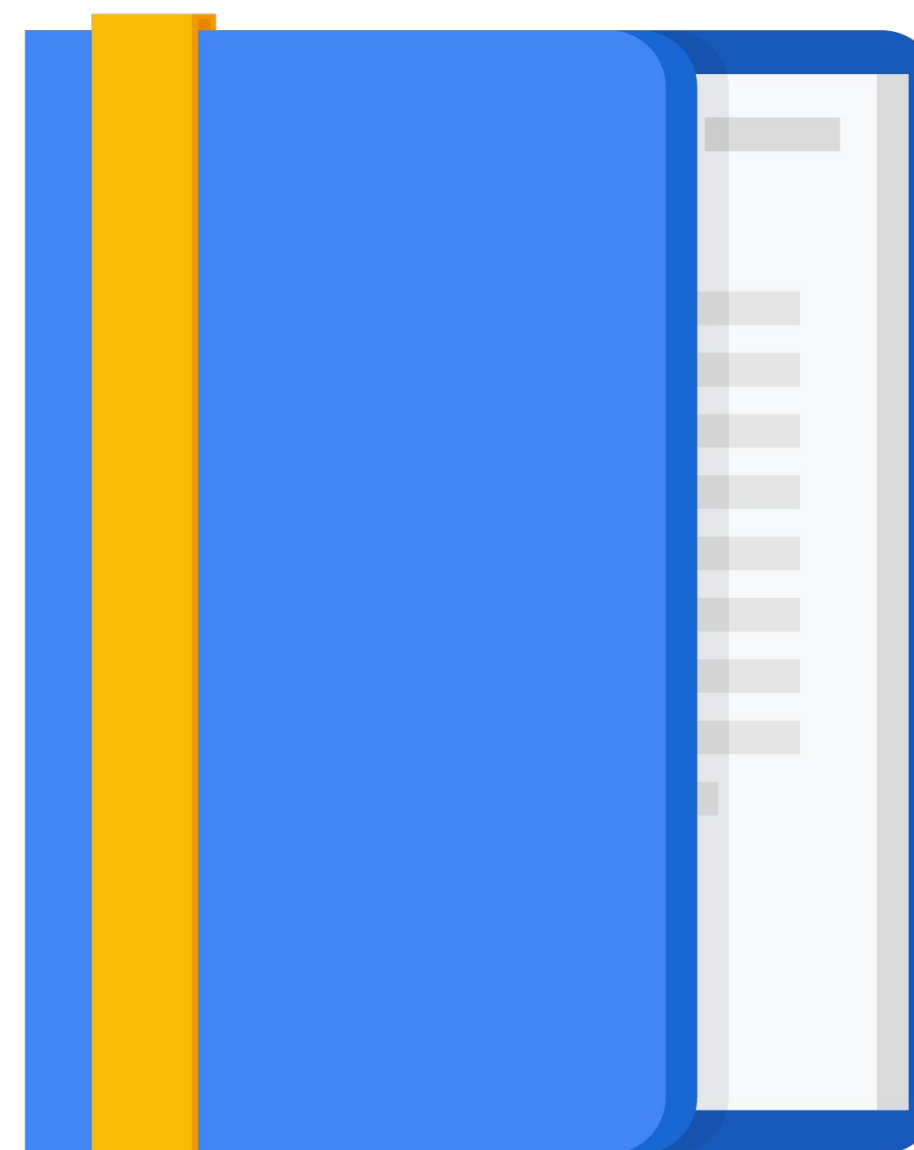
Aprenderá a realizar estas tareas:

Trabajar con el comando de kubectl.

Comprender cómo se usan los objetos Deployment en Kubernetes.

Comprender la arquitectura de red de los Pods.

Comprender las abstracciones de almacenamiento de Kubernetes.



Temario

[El comando de kubectl](#)

Objetos Deployment

Lab: Cómo crear
implementaciones de
Google Kubernetes Engine

Redes de Pods

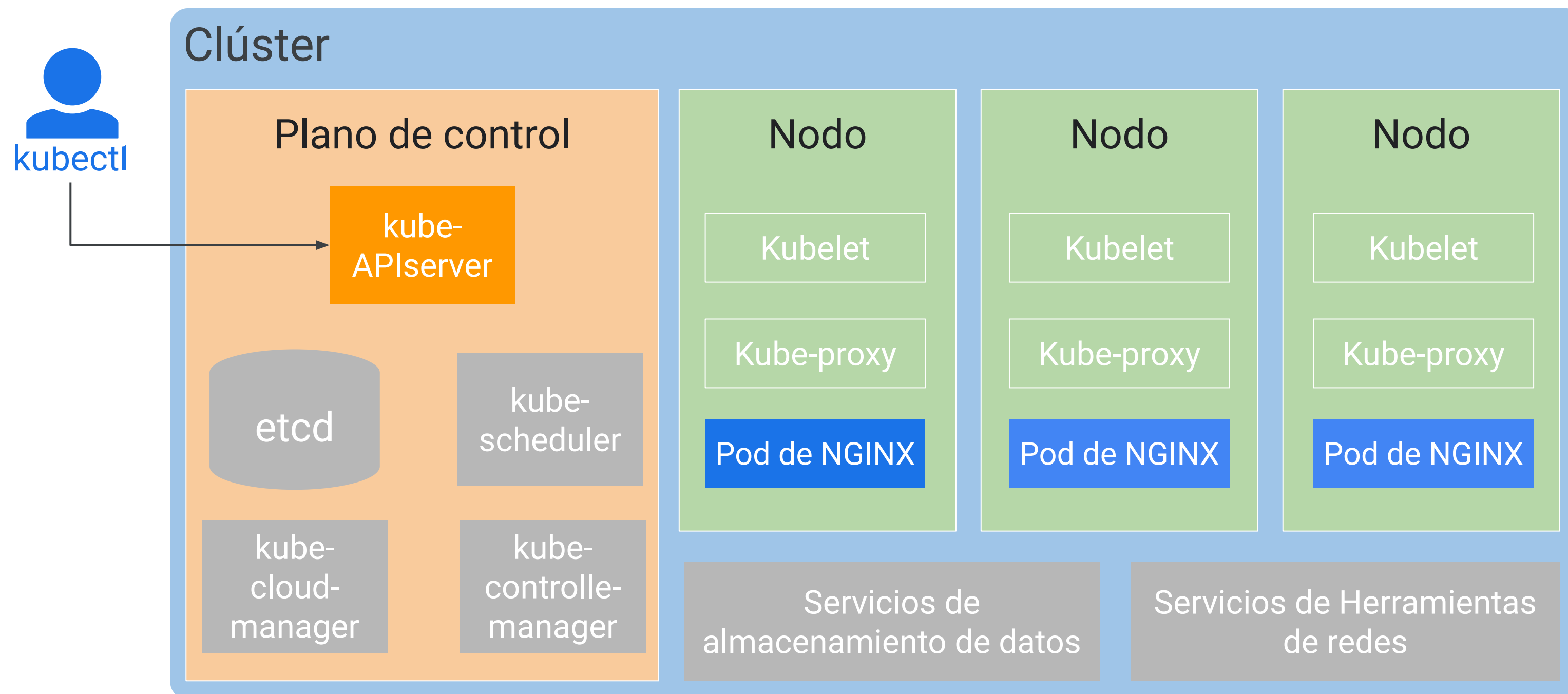
Volumes

Lab: Cómo configurar un
almacenamiento persistente para
Google Kubernetes Engine

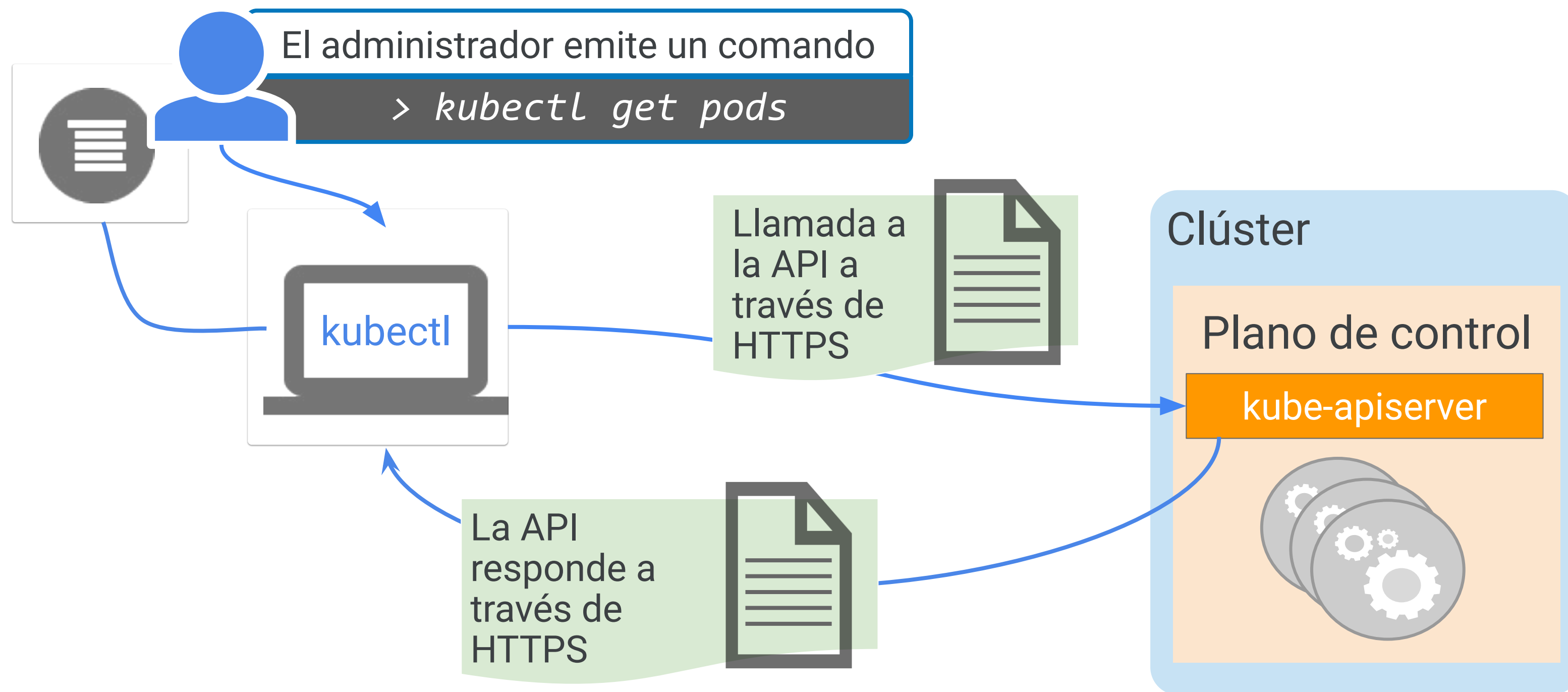
Cuestionario

Resumen

Kubectl transforma las entradas de la línea de comandos en llamadas a la API



Usa kubectl para ver una lista de Pods en un clúster



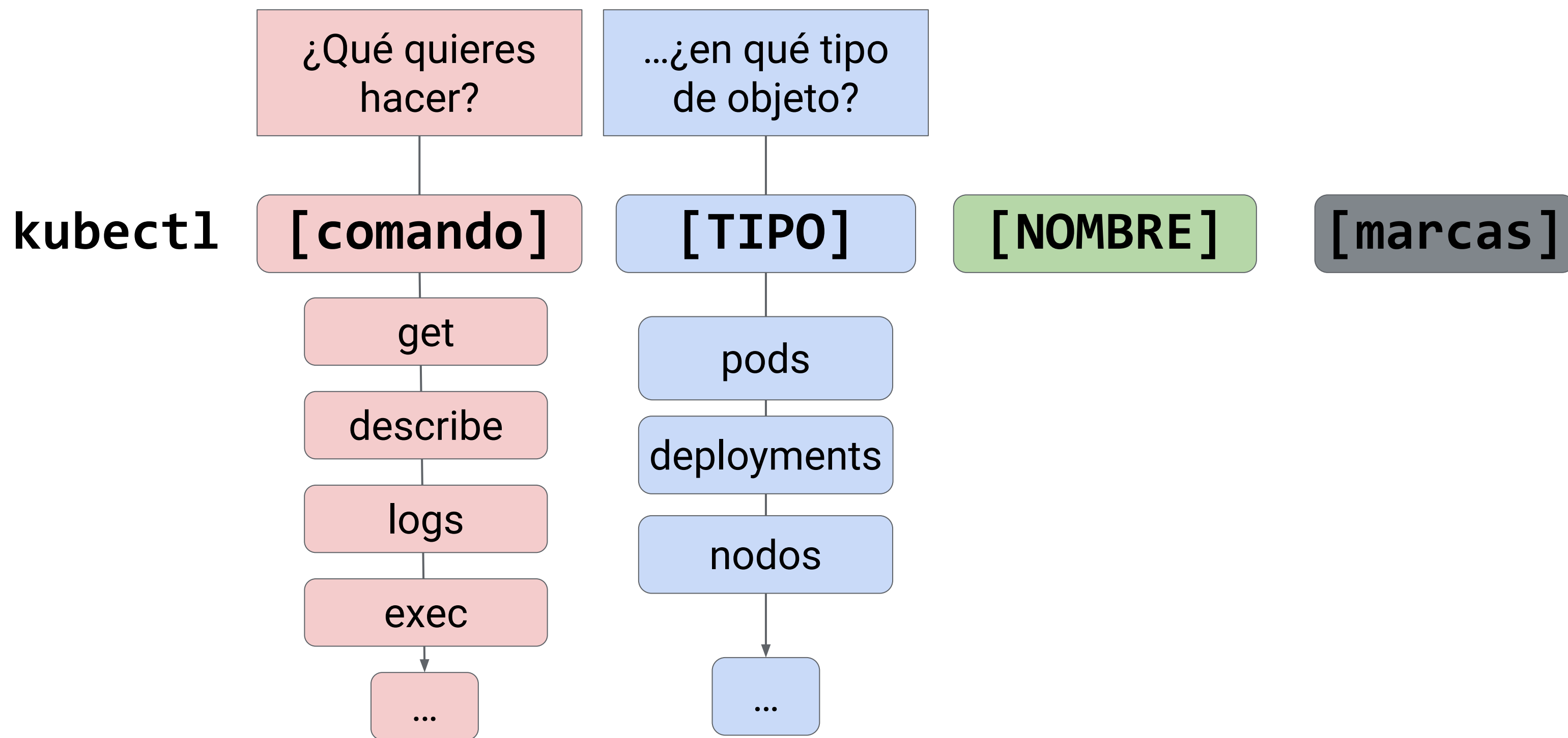
Primero se debe configurar kubectl

- Se basa en un archivo de configuración: `$HOME/.kube/config`.
- El archivo de configuración contiene lo siguiente:
 - El nombre del clúster de destino
 - Las credenciales del clúster
- Configuración actual: `kubectl config view`.
- Acceso a un Pod de forma interactiva.

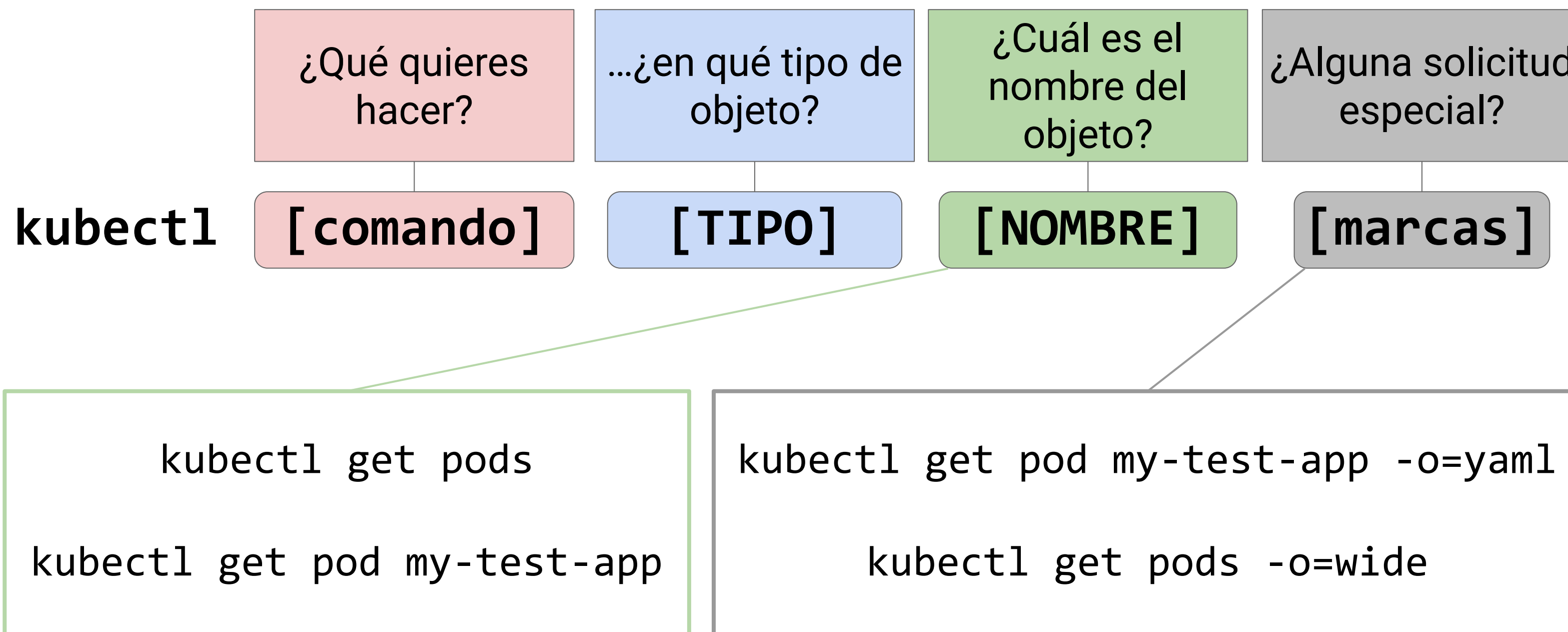
Conéctese a un clúster de Google Kubernetes Engine

```
$ gcloud container clusters \
  get-credentials [CLUSTER_NAME] \
  --zone [ZONE_NAME]
```

La sintaxis del comando de kubectl tiene varias partes



La sintaxis del comando de kubectl tiene varias partes



El comando de kubectl tiene muchos usos

- Crear objetos de Kubernetes
- Ver objetos
- Borrar objetos
- Visualizar y exportar opciones de configuración

Temario

El comando de kubectl

[Objetos Deployment](#)

Lab: Cómo crear
implementaciones de
Google Kubernetes Engine

Redes de Pods

Volumes

Lab: Cómo configurar un
almacenamiento persistente para
Google Kubernetes Engine

Cuestionario

Resumen

Los objetos Deployment declaran el estado de los Pods



Aplicación de actualizaciones a los Pods



Reversión de los Pods a la revisión anterior

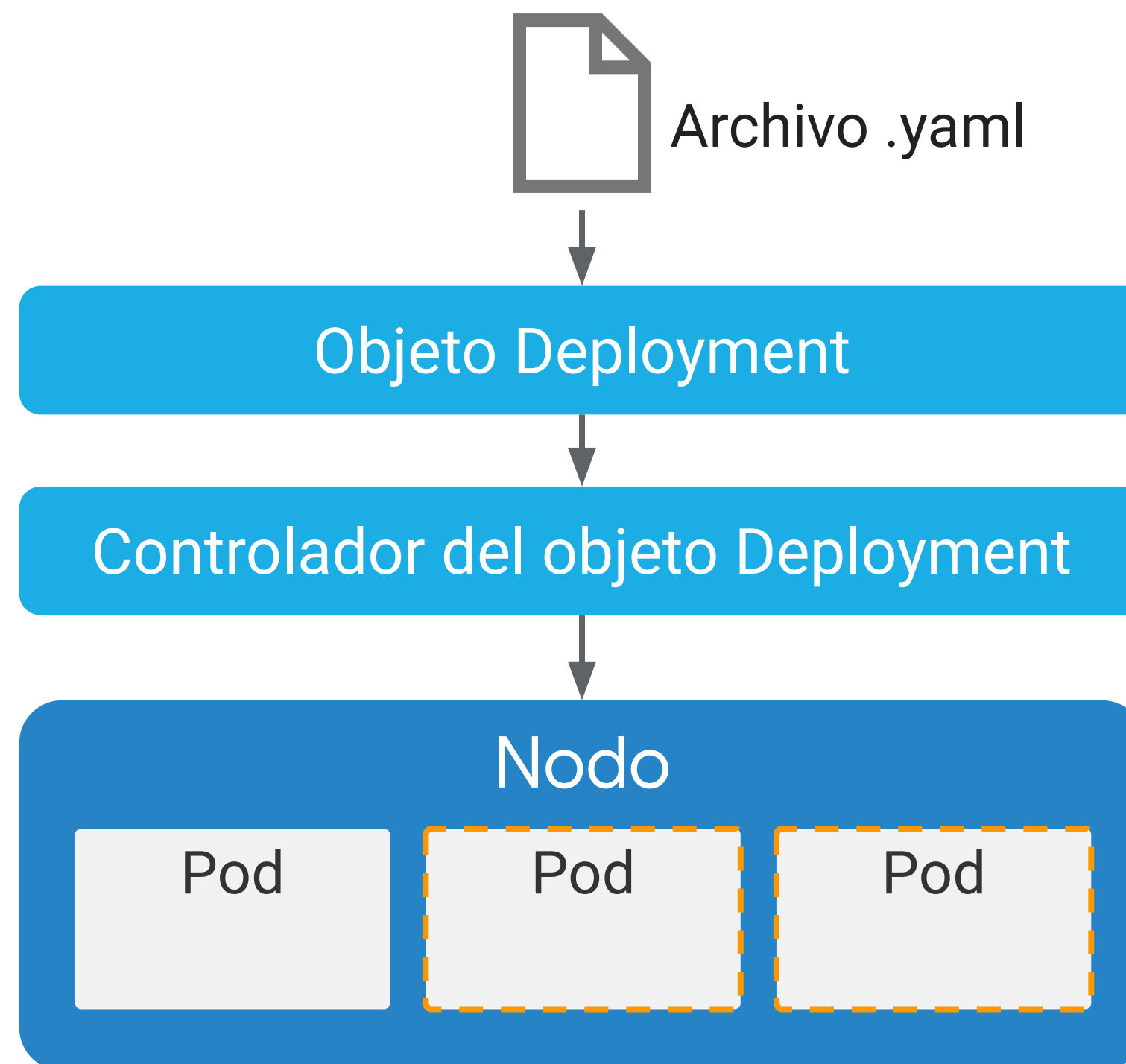


Escalamiento o ajuste de escala automático de los Pods



Ideales para aplicaciones sin estado

El proceso de los objetos Deployment consta de dos partes



Archivo de objeto Deployment en formato YAML

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: my-app
    spec:
      containers:
      - name: my-app
        image: gcr.io/demo/my-app:1.0
        ports:
        - containerPort: 8080
```

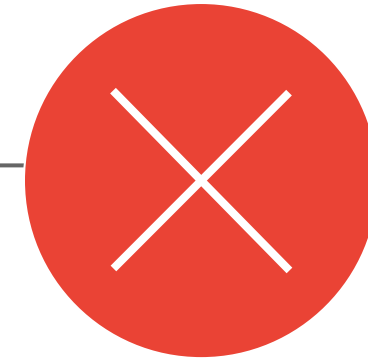
El objeto Deployment tiene tres estados distintos de ciclo de vida



Estado en
progreso



Estado
completo



Estado con
error

Existen tres formas de crear un objeto Deployment

1

```
$ kubectl apply -f [DEPLOYMENT_FILE]
```

2

```
$ kubectl create deployment \  
[DEPLOYMENT_NAME] \  
  --image [IMAGE]:[TAG] \  
  --replicas 3 \  
  --labels [KEY]=[VALUE] \  
  --port 8080 \  
  --generator deployment/apps.v1 \  
  --save-config
```


Existen tres formas de crear un objeto Deployment

3

←
Create a deployment

1
Container

Edit container

☒ Existing container image

☐ New container image

Image path *

nginx:latest

SELECT

Enter your image path, or choose from Google Container Registry. You can also try to deploy with official nginx image nginx:latest.

Environment variables

+ ADD ENVIRONMENT VARIABLE

Initial command

Overrides the default entrypoint of the container image.

CANCEL
DONE

ADD CONTAINER

CONTINUE

←
Create a deployment

✓
Container

2
Configuration

A deployment is a configuration which defines how Kubernetes deploys, manages, and scales your container image. Kubernetes will ensure your system matches this configuration.

Application name *

nginx-1

Namespace *

default

Labels

Key *	Value
app	nginx-1

+ ADD KUBERNETES LABEL

Configuration YAML

Kubernetes deployments are defined declaratively using YAML files. The best practice is to store these files in version control, so you can track changes to your deployment configuration over time.

VIEW YAML

Cluster

Kubernetes Cluster

standard-cluster-1 (us-central1-a)

▼

Cluster in which the deployment will be created.

CREATE NEW CLUSTER

DEPLOY

Google Cloud

Usa `kubectl` para inspeccionar tu objeto Deployment, o bien obtén el resultado de la configuración del objeto Deployment en formato YAML

```
$ kubectl get deployment [DEPLOYMENT_NAME]
```

```
master $ kubectl get deployment nginx-deployment
NAME           DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment 3         3         3            3           3m
```

```
$ kubectl get deployment [DEPLOYMENT_NAME] -o yaml > this.yaml
```

Use el comando “describe” para obtener información detallada

```
$ kubectl describe deployment [DEPLOYMENT_NAME]
```

```
master $ kubectl describe deployment nginx-deployment
Name:                nginx-deployment
Namespace:            default
CreationTimestamp:    Fri, 12 Oct 2018 15:23:46 +0000
Labels:               app=nginx
Annotations:          deployment.kubernetes.io/revision=1
Selector:             app=nginx
Replicas:             3 desired | 3 updated | 3 total | 3 available | 0 unavailable
StrategyType:         RollingUpdate
MinReadySeconds:      0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
    nginx:
      Image:          nginx:1.15.4
      Port:           80/TCP
      Host Port:      0/TCP
```

O usa la consola de Cloud

✓ nginx-deployment

1

To let others access your deployment, expose it to create a service

OVERVIEW

DETAILS

REVISION HISTORY

EVENTS

LOGS

YAML

CPU ?

Memory ?

0.002

0.001

0

UTC+1

12:30 PM

12:40 PM

12:50 PM

1:00 PM

1:10 PM

UTC+1

12:30 PM

Cluster

standard-cluster-1

Namespace

default

Labels

app: nginx

Logs ?

Container logs, Audit logs

Replicas

3 updated, 3 ready, 3 available, 0 unavailable

Pod specification

Revision 1, containers: [nginx](#)

Active revisions

Revision	Name	Status	Summary	Created on	Pods running/total
1	nginx-deployment-5d59d67564	✓ OK	nginx: nginx:1.7.9	Oct 13, 2021, 1:16:02 PM	3/3

Managed pods

Revision	Name	Status	Restarts	Created on
1	nginx-deployment-5d59d67564-2dsfb	✓ Running	0	Oct 13, 2021, 1:16:02 PM
1	nginx-deployment-5d59d67564-8cknj	✓ Running	0	Oct 13, 2021, 1:16:02 PM
1	nginx-deployment-5d59d67564-d898b	✓ Running	0	Oct 13, 2021, 1:16:02 PM

✓ nginx-deployment

1

To let others access your deployment, expose it to create a service

OVERVIEW

DETAILS

REVISION HISTORY

EVENTS

LOGS

YAML

Cluster

standard-cluster-1

Namespace

default

Created

Oct 13, 2021, 1:16:02 PM

Labels

app: nginx

Annotations

deployment.kubernetes.io/revision: 1

SHOW ALL ANNOTATIONS

Replicas

3 updated, 3 ready, 3 available, 0 unavailable

Label selector

app = nginx

Update strategy ?

Rolling update, Max unavailable: 25%, Max surge: 25%

Min time ready before available

0 s

Progress deadline

600 s

Revision history limit

10

Pod specification

Revision 1

Labels

app: nginx

Termination grace period

30

Restart policy

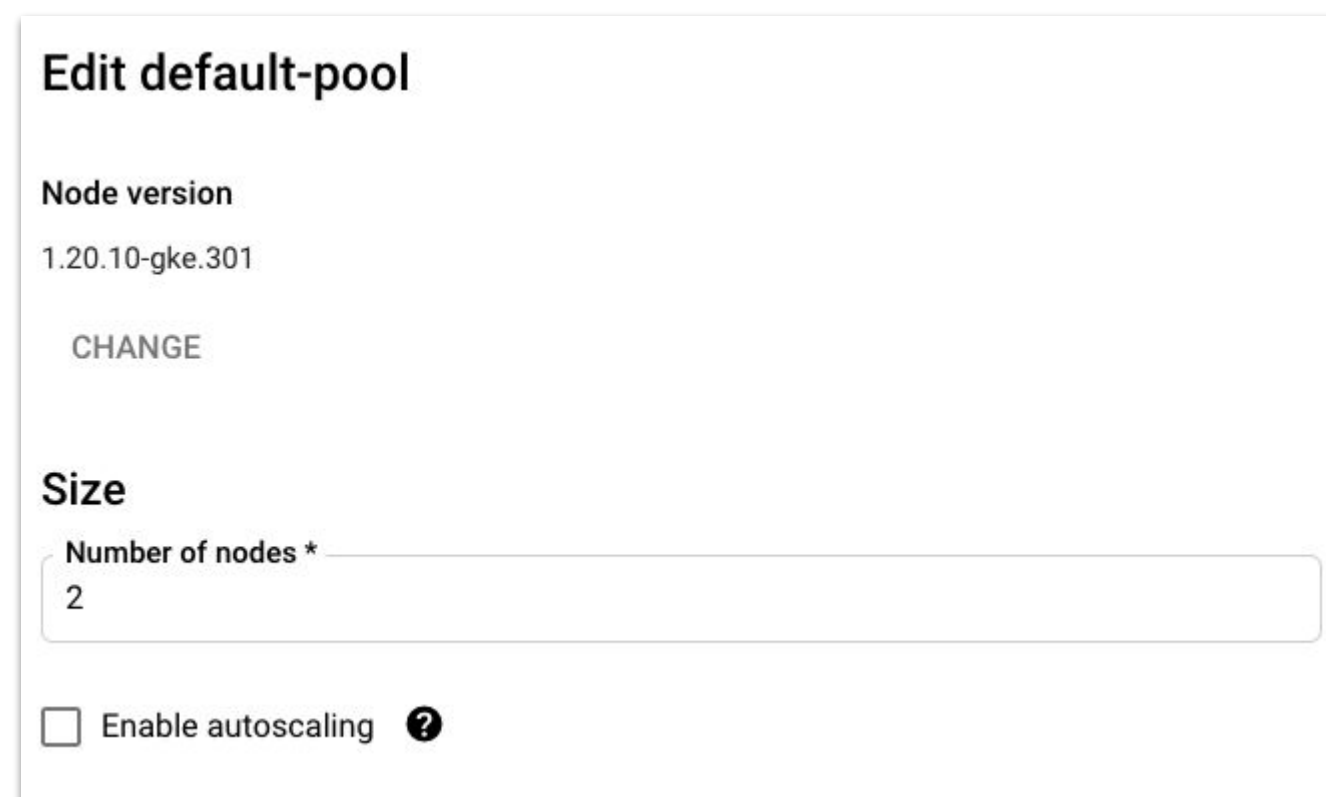
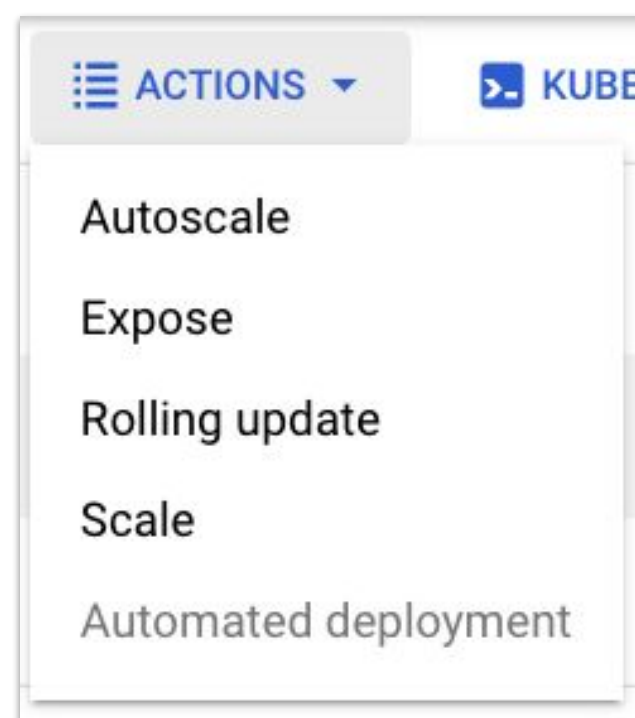
Always

Containers

nginx

Puede escalar el objeto Deployment de forma manual

```
$ kubectl scale deployment  
[DEPLOYMENT_NAME] --replicas=5
```



También puede escalar automáticamente el objeto Deployment

```
$ kubectl autoscale deployment [DEPLOYMENT_NAME] --min=1 --max=3  
--cpu-percent=80
```

Autoscale

Automatically scale the number of pods.

Minimum number of Pods (Optional)

Maximum number of Pods

Target CPU utilization in percent (Optional)

[CANCEL](#) [DISABLE AUTOSCALER](#) [AUTOSCALE](#)

Puedes actualizar un objeto Deployment de diferentes maneras

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-app
spec:
  replicas: 3
  template:
    spec:
      containers:
        - name: my-app
          image: gcr.io/demo/my-app:1.0
          ports:
            - containerPort: 8080
```

```
$ kubectl apply -f [DEPLOYMENT_FILE]
```

```
$ kubectl set image deployment
[DEPLOYMENT_NAME] [IMAGE] [IMAGE]:[TAG]
```

```
$ kubectl edit \
  deployment/[DEPLOYMENT_NAME]
```

Puedes actualizar un objeto Deployment de diferentes maneras

[REFRESH](#) [EDIT](#) [DELETE](#) [ACTIONS](#) [KUBECTL](#)

Rolling update

Update workload Pods to a new application version.

Minimum seconds ready
0

Maximum surge
25%

Maximum unavailable
25%

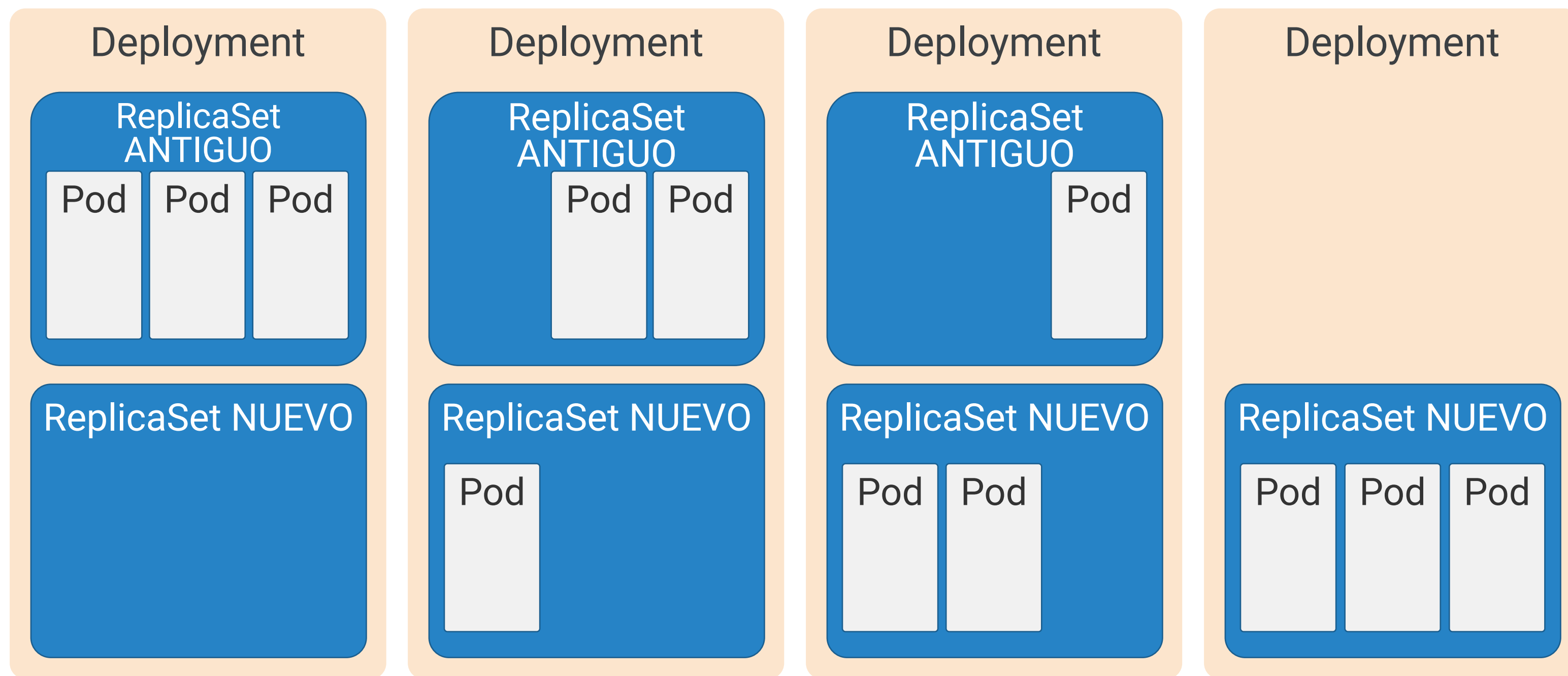
Container images

Image of nginx *
nginx:1.7.9

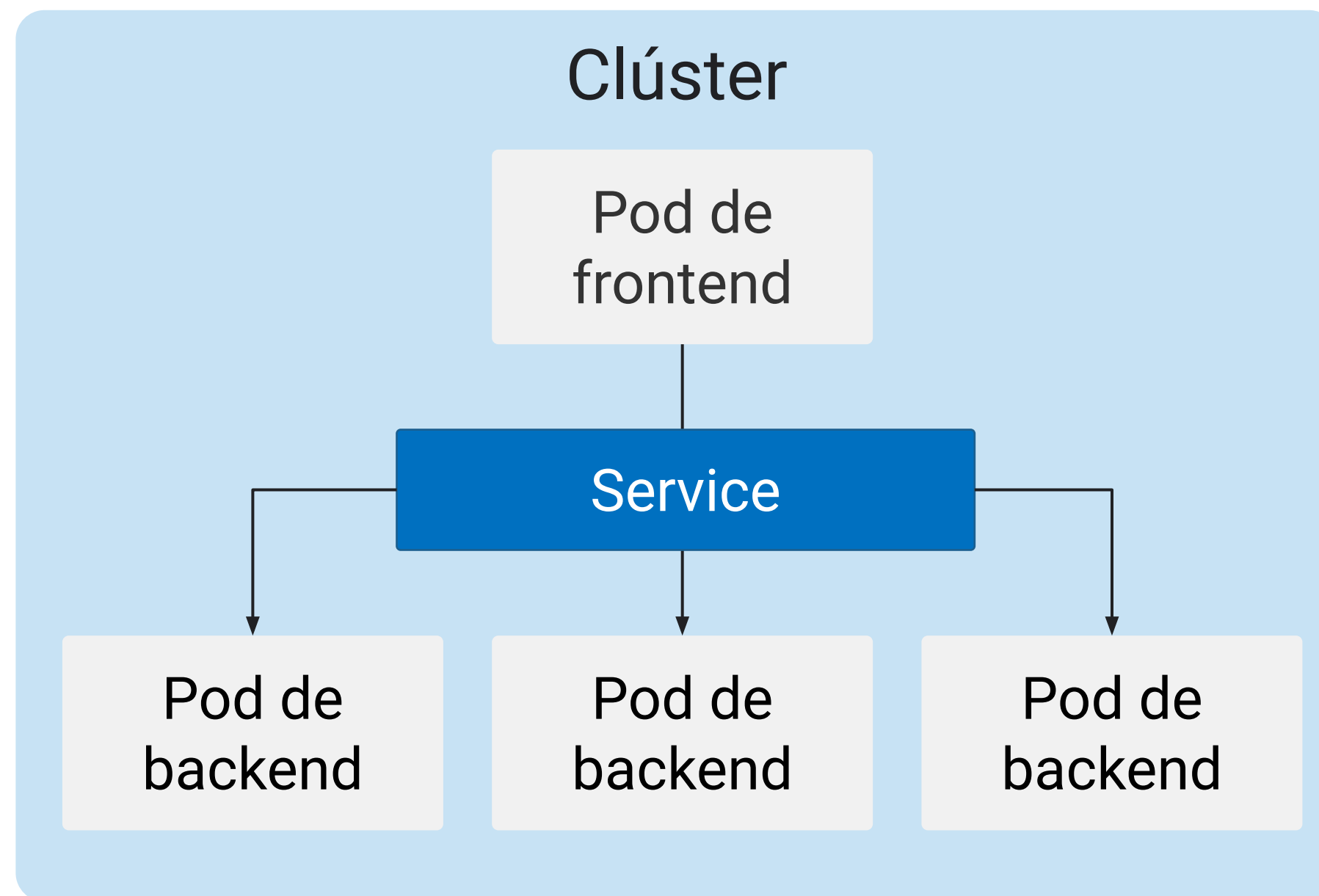
* Indicates required field

[CANCEL](#) [UPDATE](#)

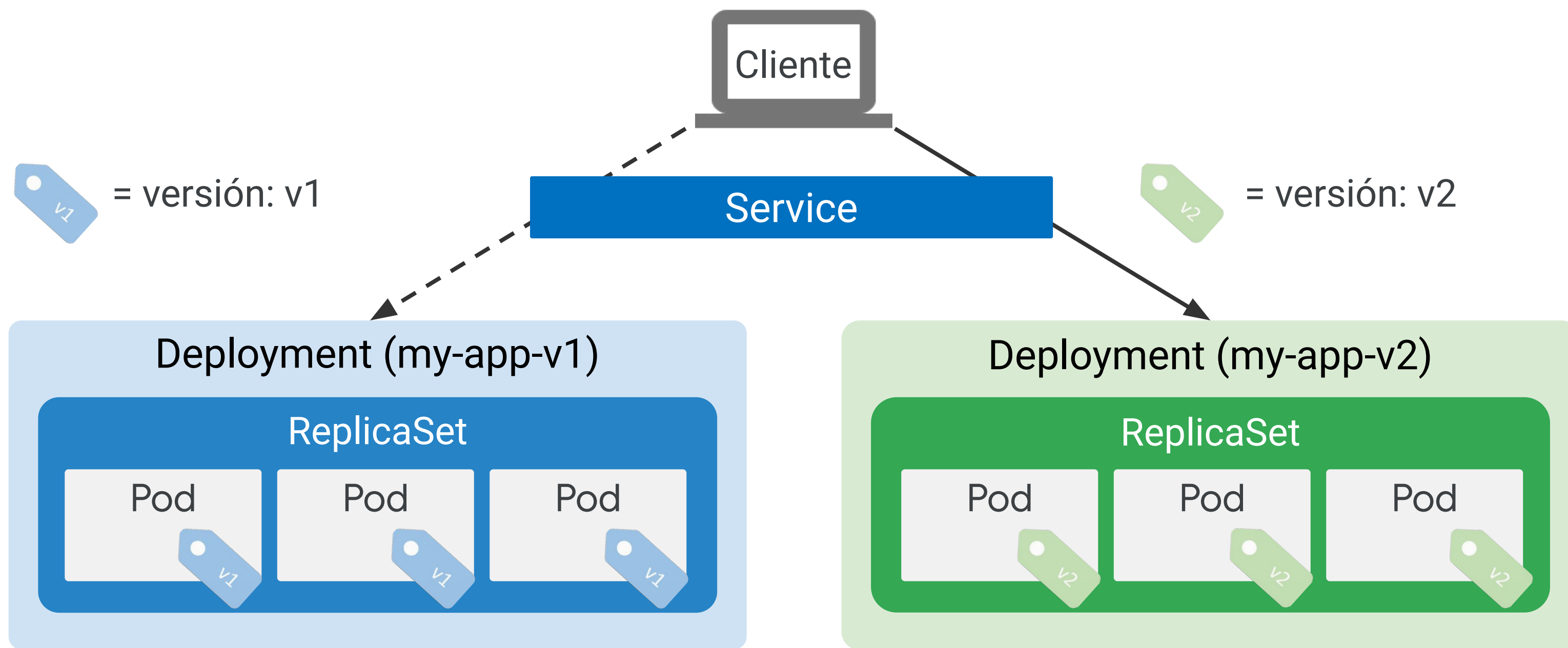
El proceso que conlleva la actualización de un objeto Deployment



El Service es una representación de red estable de un conjunto de Pods



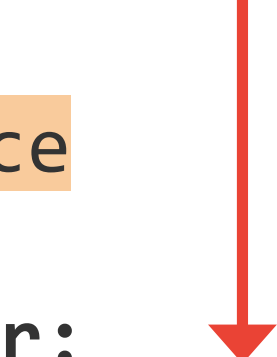
Una estrategia de implementación azul-verde garantiza que los servicios de la app permanezcan disponibles



Cómo aplicar una estrategia de implementación azul-verde

```
[...]  
kind: Service  
spec:  
  selector:  
    app: my-app  
    version: v1  
[...]
```

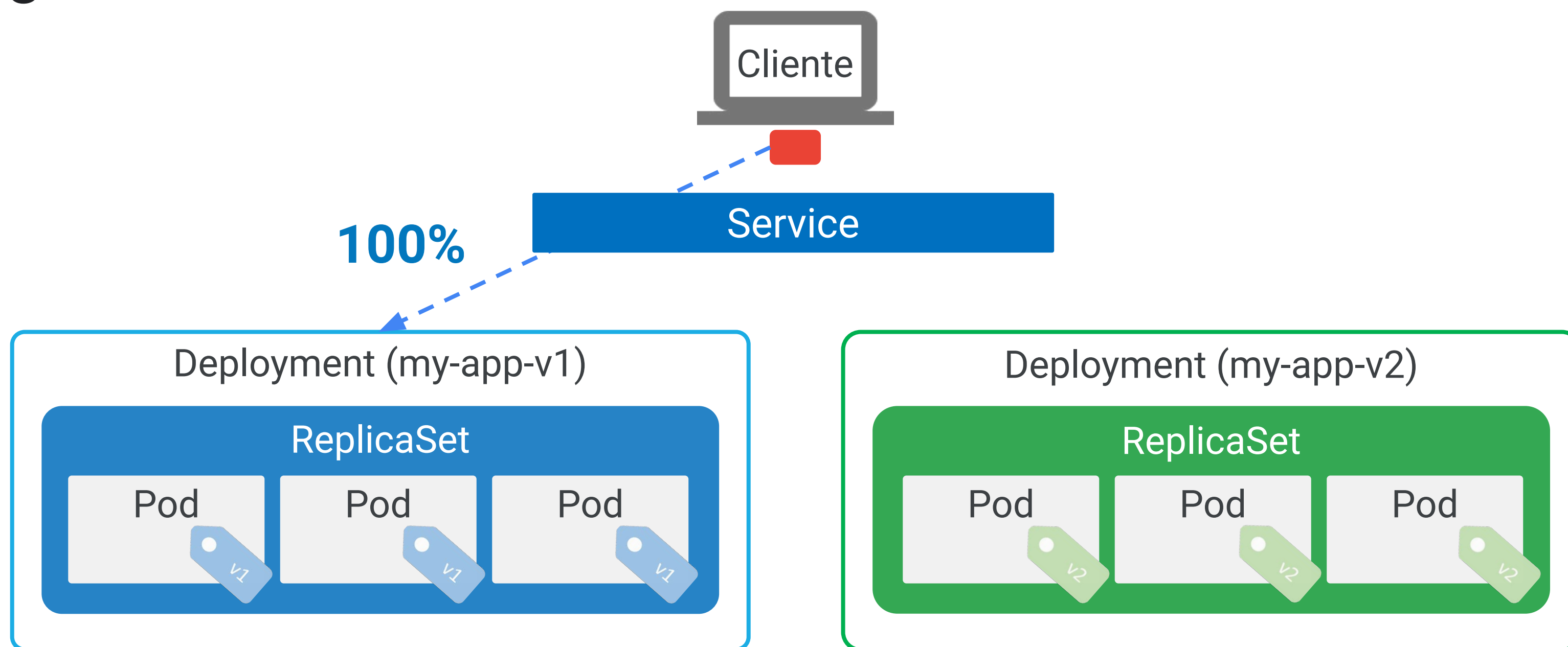
```
[...]  
kind: Service  
spec:  
  selector:  
    app: my-app  
    version: v2  
[...]
```



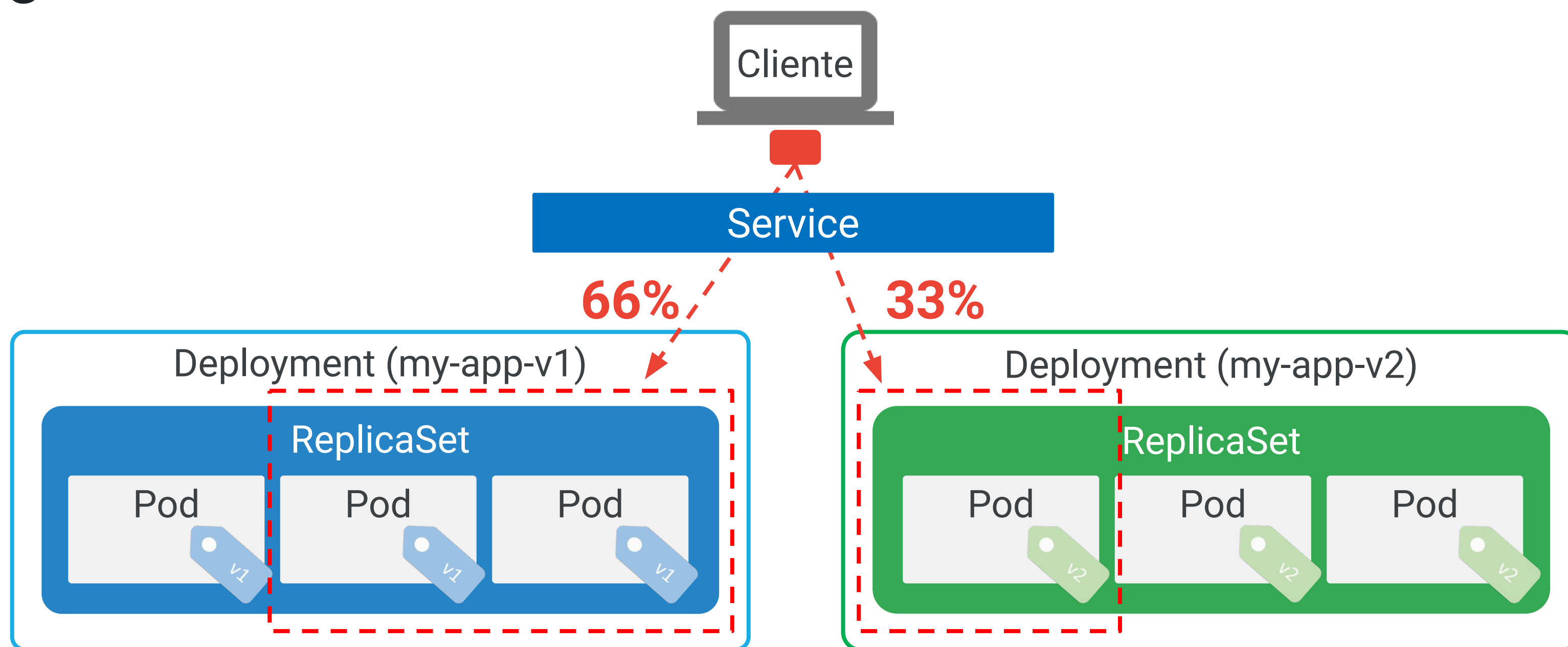
```
$ kubectl apply -f my-app-v2.yaml
```

```
$ kubectl patch service my-app-service -p \  
'{"spec":{"selector":{"version":"v2"}}}'
```

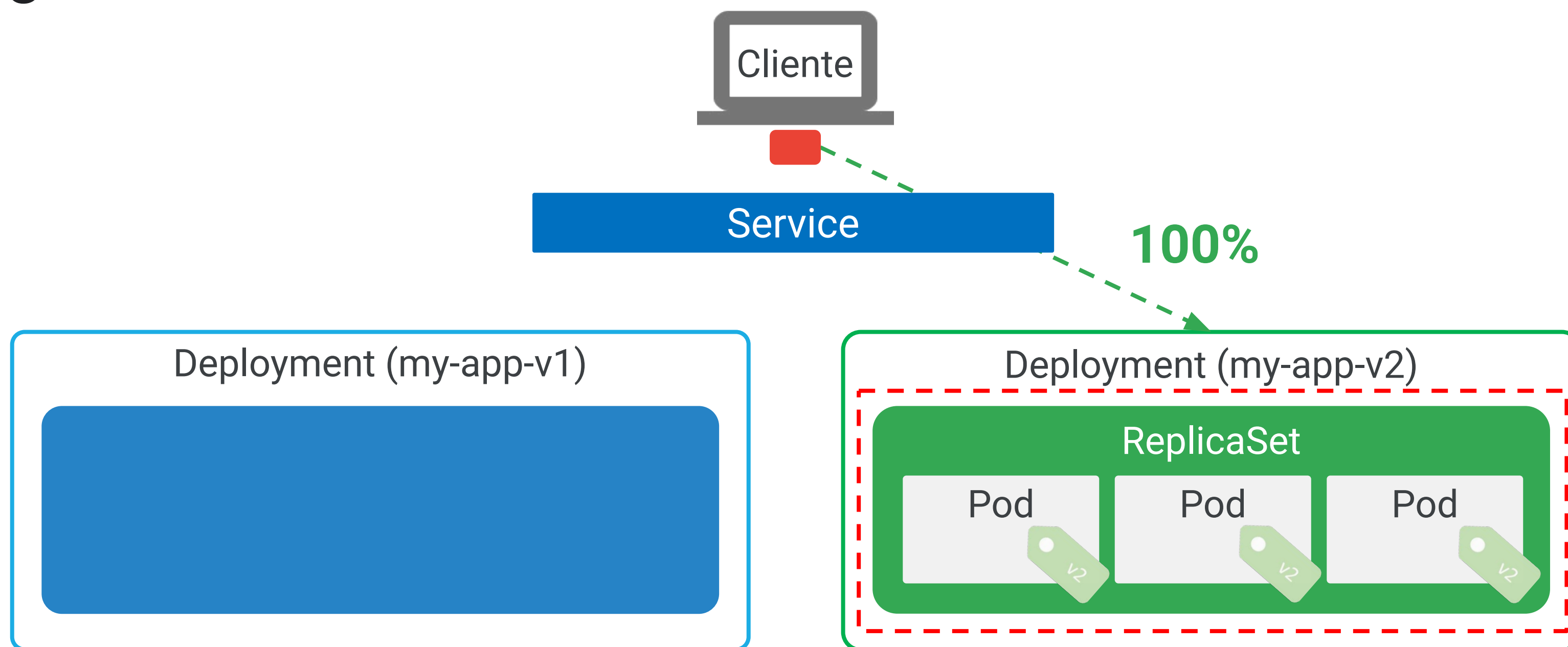
La implementación de versiones canary es una estrategia de actualización en la que el tráfico se cambia de forma gradual a la versión nueva



La implementación de versiones canary es una estrategia de actualización en la que el tráfico se cambia de forma gradual a la versión nueva



La implementación de versiones canary es una estrategia de actualización en la que el tráfico se cambia de forma gradual a la versión nueva



Cómo aplicar una implementación de versiones canary

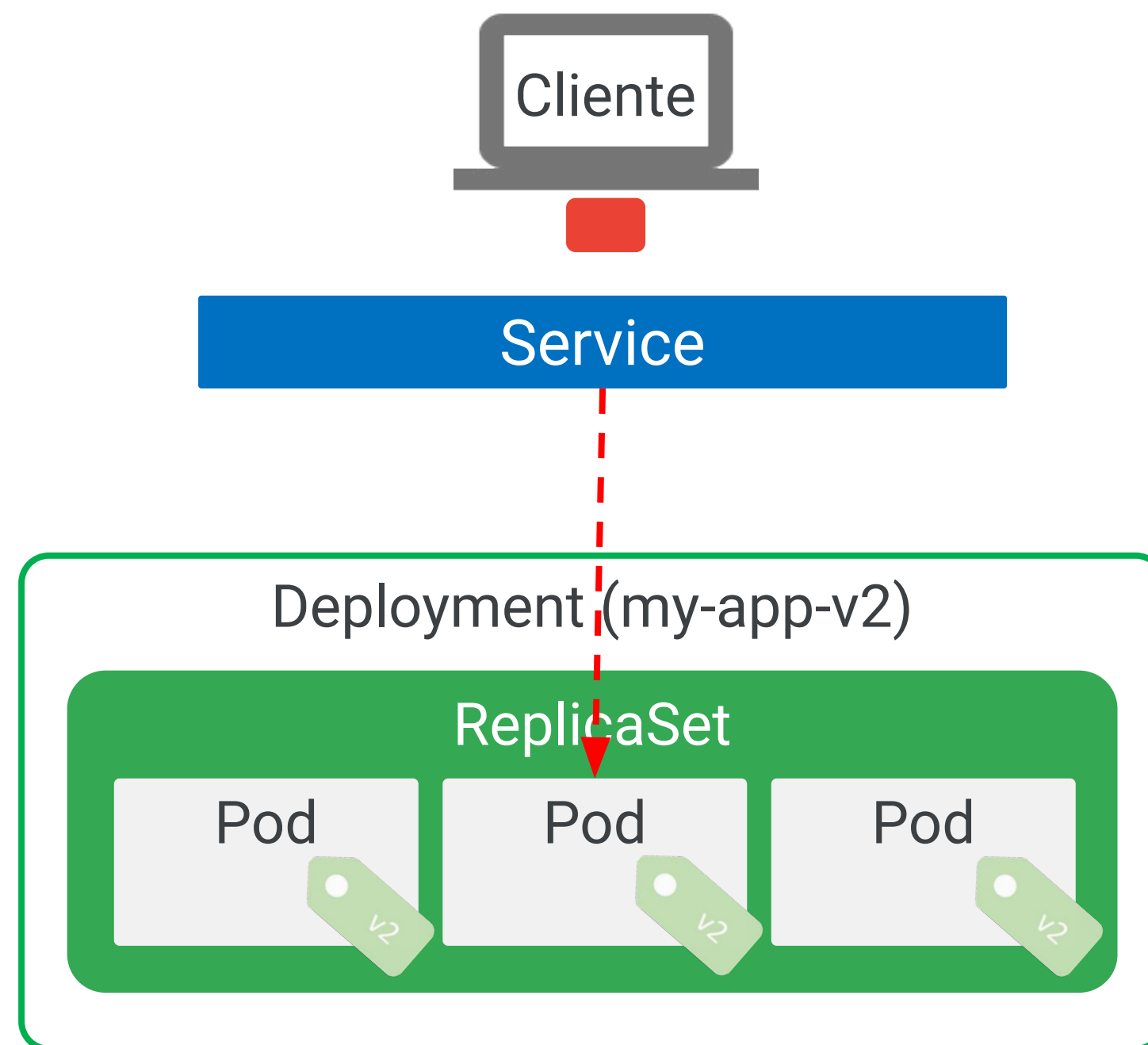
```
[...]  
kind: Service  
spec:  
  selector:  
    app: my-app  
[...]
```

```
$ kubectl apply -f my-app-v2.yaml
```

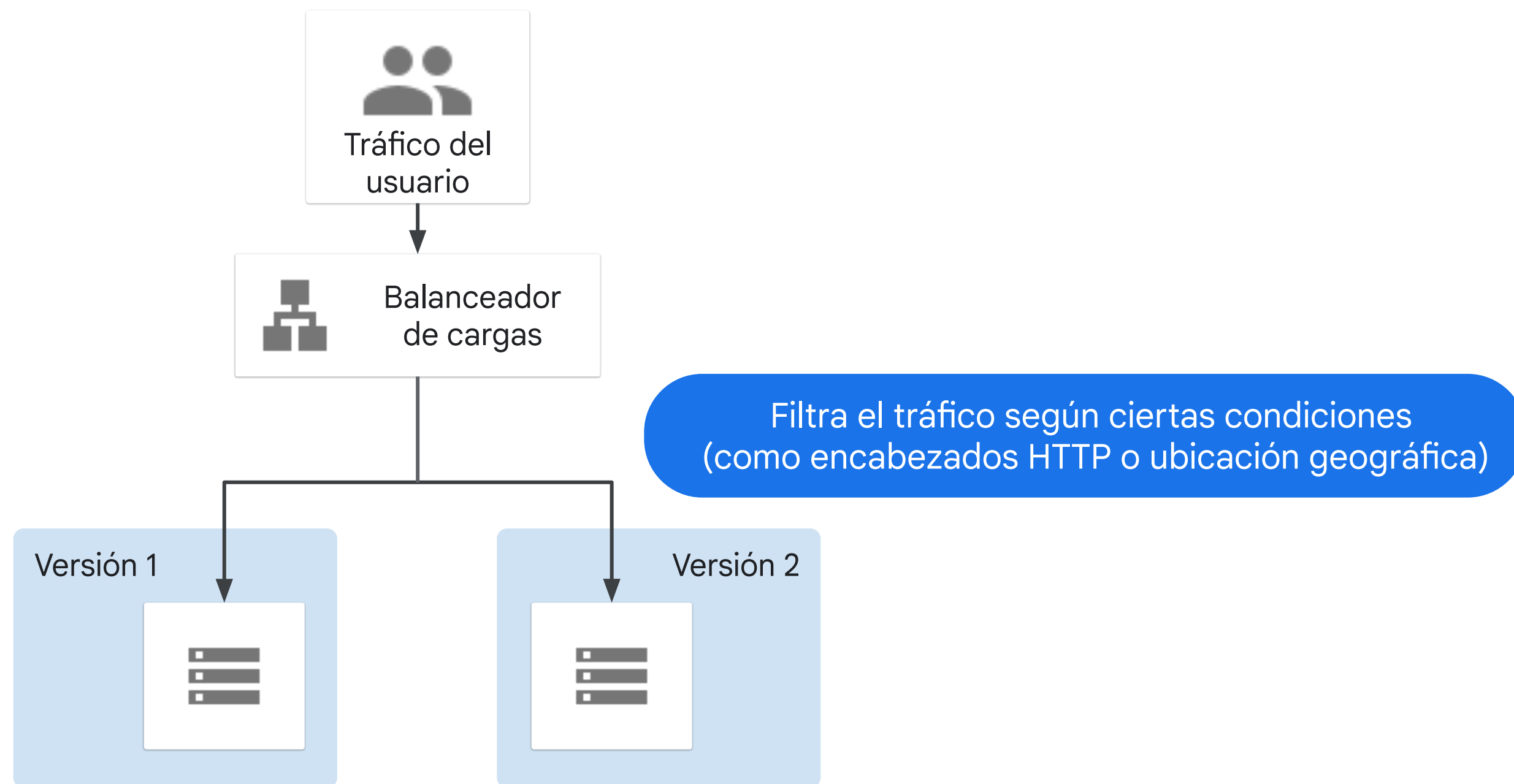
```
$ kubectl scale deploy/my-app-v2 --replicas=10
```

```
$ kubectl delete -f my-app-v1.yaml
```

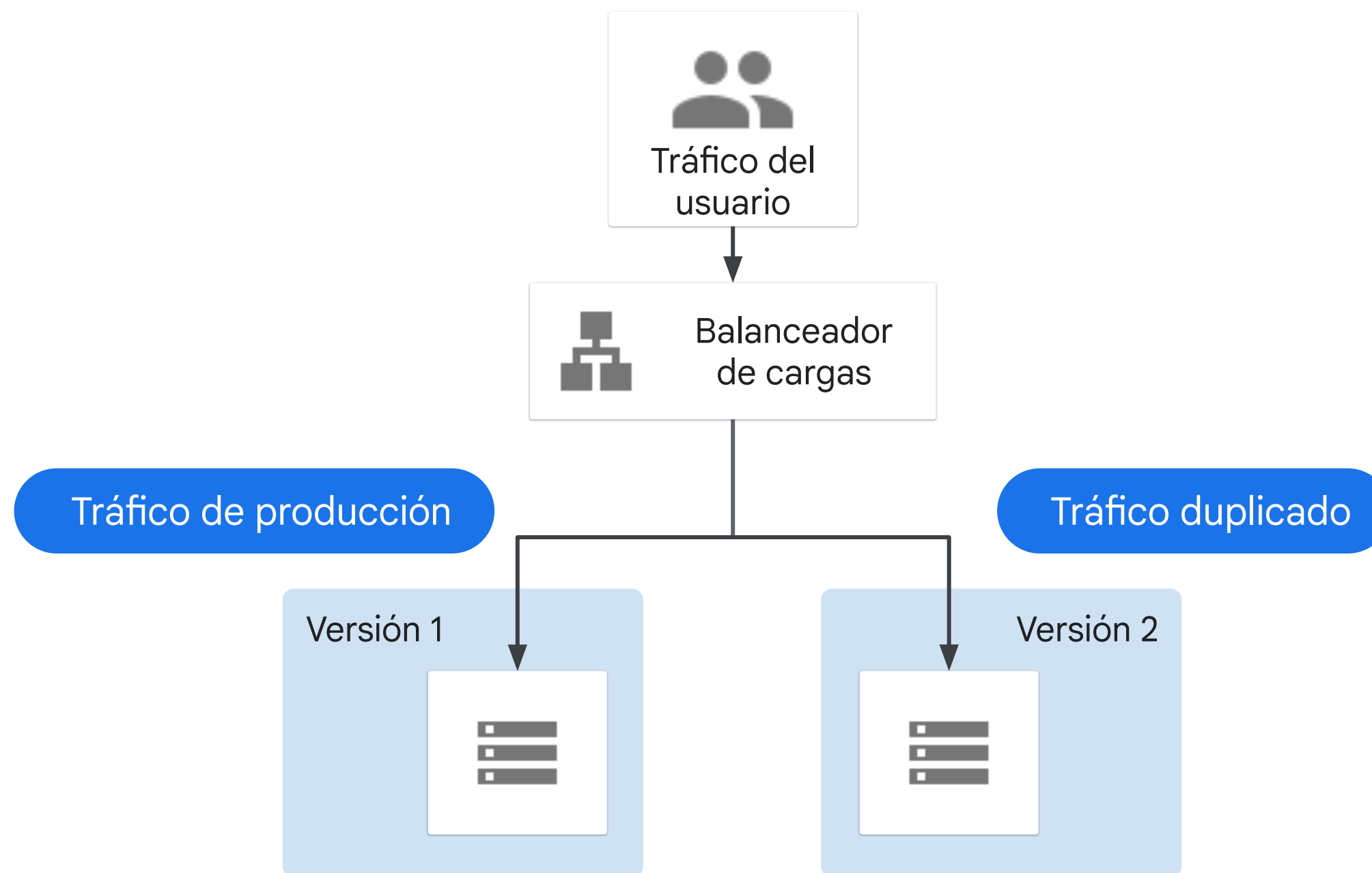

La afinidad de sesión garantiza que todas las solicitudes de clientes se envíen al mismo Pod



Las pruebas A/B sirven para medir la eficacia de una funcionalidad en una aplicación



Las pruebas paralelas te permiten ejecutar una versión nueva y oculta



Cómo elegir la estrategia adecuada

Patrón de implementación o de prueba	Sin tiempo de inactividad	Pruebas de tráfico de producción real	Lanzamiento a los usuarios según las condiciones	Duración de la reversión	Impacto en el hardware y los costos de la nube
Recreación Se cancela la versión 1 y se lanza la versión 2.	✗	✗	✗	Rápida pero perjudicial debido al tiempo de inactividad	No se requiere ninguna configuración adicional
Actualización progresiva La versión 2 se lanza de forma gradual y reemplaza a la versión 1.	✓	✗	✗	Lenta	Puede requerir configuración adicional para actualizaciones de aumento
Azul-verde La versión 2 se lanza junto con la versión 1. El tráfico se pasa a la versión 2 después de la prueba.	✓	✗	✗	Instantánea	Se deben mantener los entornos azul y verde en simultáneo
Versión canary La versión 2 se lanza a un subconjunto de usuarios y, luego, se realiza un lanzamiento completo.	✓	✓	✗	Rápida	No se requiere ninguna configuración adicional
A/B La versión 2 se lanza, en condiciones específicas a un subconjunto de usuarios.	✓	✓	✓	Rápida	No se requiere ninguna configuración adicional
Paralela La versión 2 recibe tráfico real sin afectar las solicitudes de los usuarios.	✓	✓	✗	No aplica	Se deben mantener entornos paralelos para capturar y volver a reproducir las solicitudes de los usuarios

Cómo revertir un objeto Deployment

```
$ kubectl rollout undo deployment [DEPLOYMENT_NAME]
```

```
$ kubectl rollout undo deployment [DEPLOYMENT_NAME] --to-revision=2
```

```
$ kubectl rollout history deployment [DEPLOYMENT_NAME] --revision=2
```

Política de limpieza:

- Predeterminado: 10 revisiones
- Para cambiar: `.spec.revisionHistoryLimit`

Se pueden aplicar distintas acciones a un objeto Deployment

Pausar

```
$ kubectl rollout pause deployment [DEPLOYMENT_NAME]
```

Reanudar

```
$ kubectl rollout resume deployment [DEPLOYMENT_NAME]
```

Supervisar

```
$ kubectl rollout status deployment [DEPLOYMENT_NAME]
```

Borre una implementación

```
$ kubectl delete deployment [DEPLOYMENT_NAME]
```

The screenshot shows the Google Cloud Workloads console. At the top, there are tabs for 'Workloads', 'REFRESH', 'DEPLOY', and 'DELETE'. Below these are filters for 'Cluster' and 'Namespace', along with 'RESET' and 'SAVE' buttons. A description states: 'Workloads are deployable units of computing that can be created and managed in a cluster.' Below this are tabs for 'OVERVIEW', 'COST OPTIMIZATION', and 'PREVIEW'. A filter bar shows 'Filter' with a dropdown set to 'Is system object : False' and a 'Filter workloads' button. A table lists the workloads:

<input checked="" type="checkbox"/>	Name ↑	Status	Type	Pods	Namespace	Cluster
<input checked="" type="checkbox"/>	nginx-deployment	OK	Deployment	3/3	default	standard-cluster-1

A modal dialog titled 'Delete resources' is open in the bottom right. It asks 'Are you sure you want to delete the selected resource?' and has a checked checkbox for 'Delete Horizontal Pod Autoscaler associated with selected Deployment'. At the bottom of the modal are 'CANCEL' and 'DELETE' buttons.

Temario

El comando de kubectl

Objetos Deployment

Lab: Cómo crear
implementaciones de
Google Kubernetes Engine

Redes de Pods

Volumes

Lab: Cómo configurar un
almacenamiento persistente para
Google Kubernetes Engine

Cuestionario

Resumen

Introducción al lab

Cómo crear implementaciones de
Google Kubernetes Engine

