



Introducción a las cargas de trabajo de Kubernetes



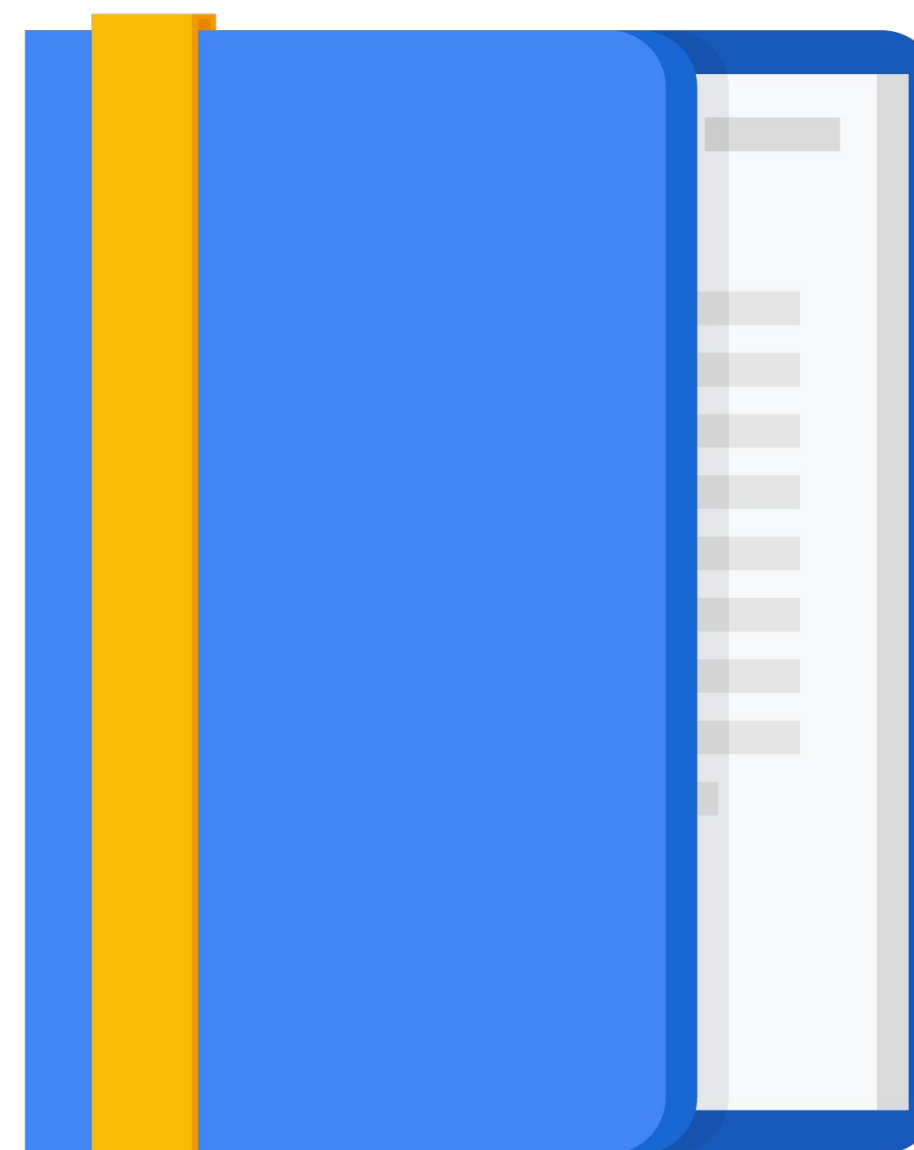
Aprenderá a realizar estas tareas:

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

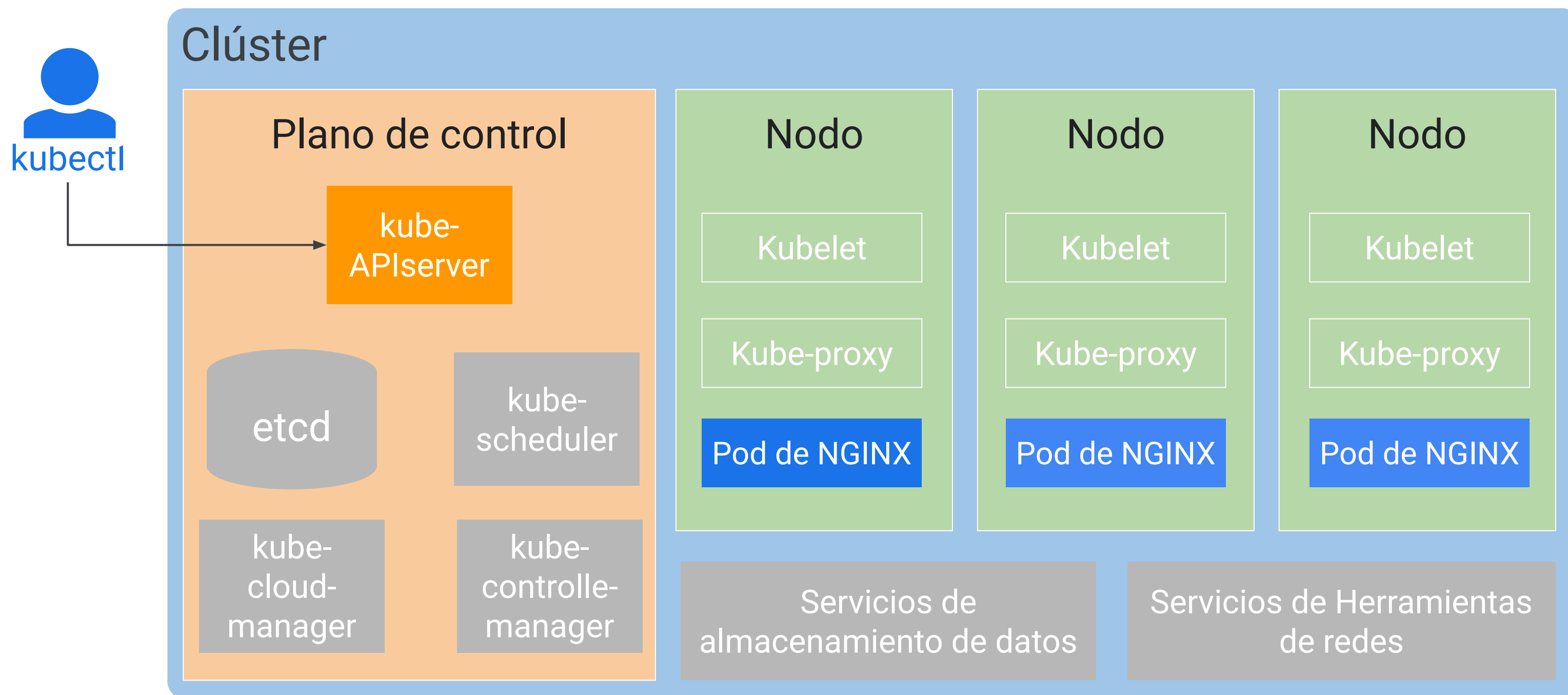
Volumes

Lab: Cómo configurar un
almacenamiento persistente para
Google Kubernetes Engine

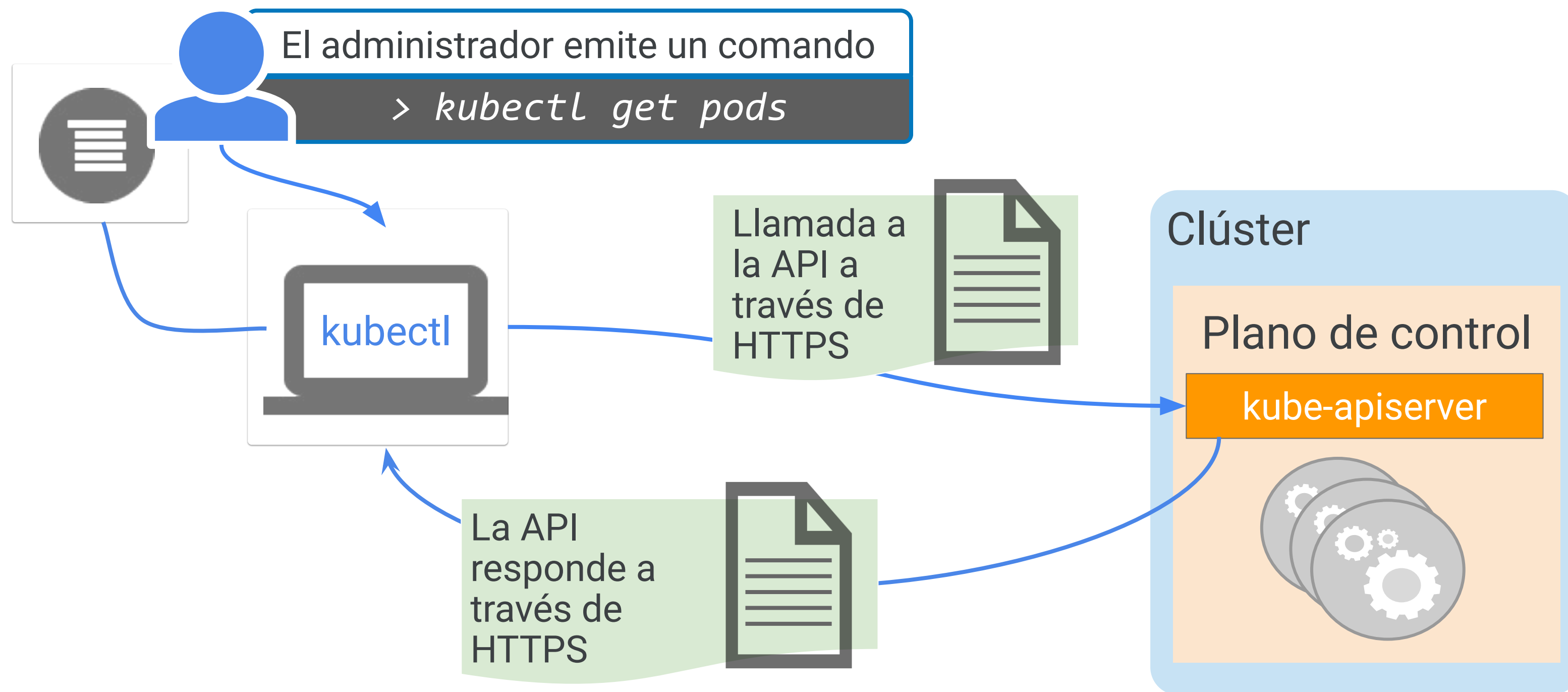
Cuestionario

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



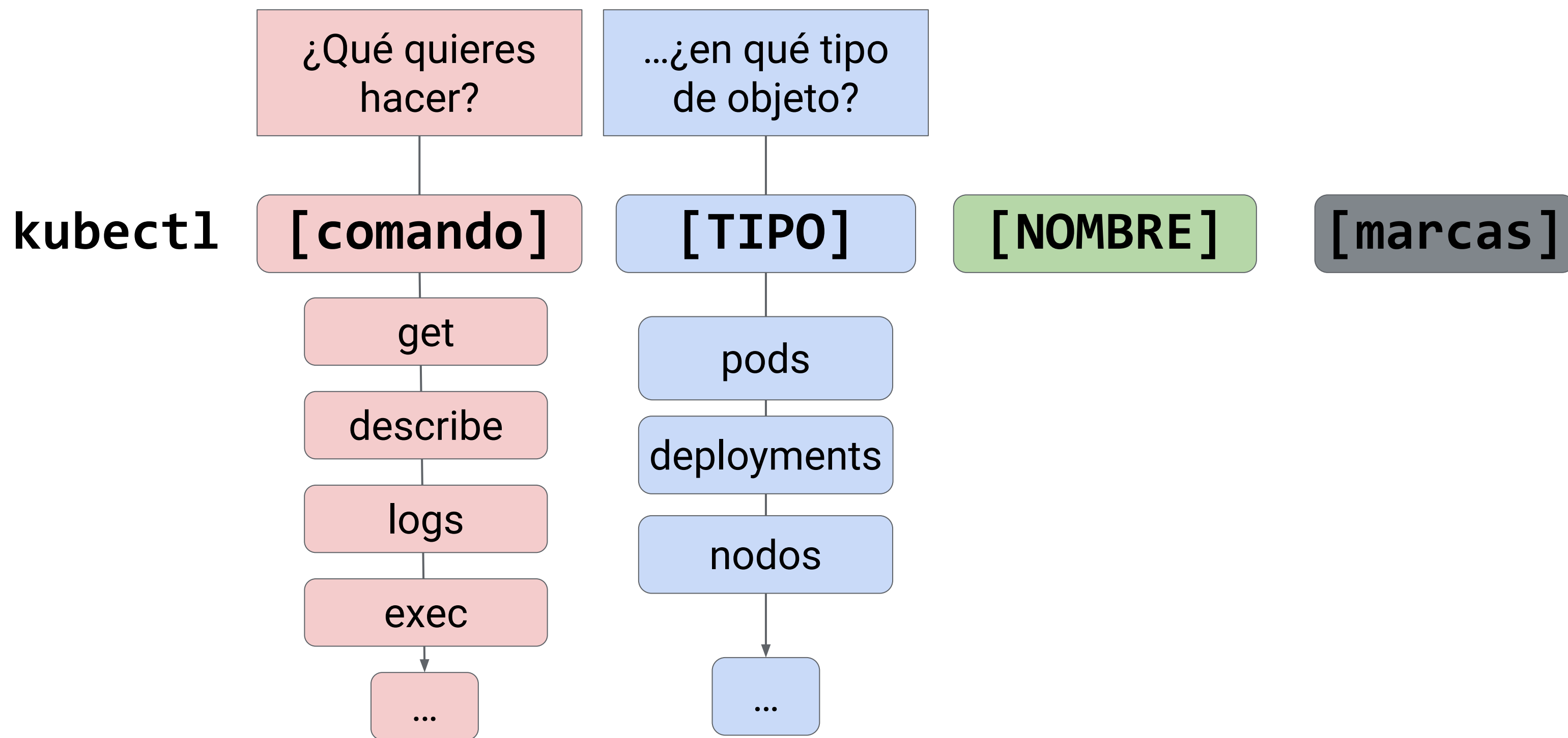
Primero se debe configurar kubectl

- 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.

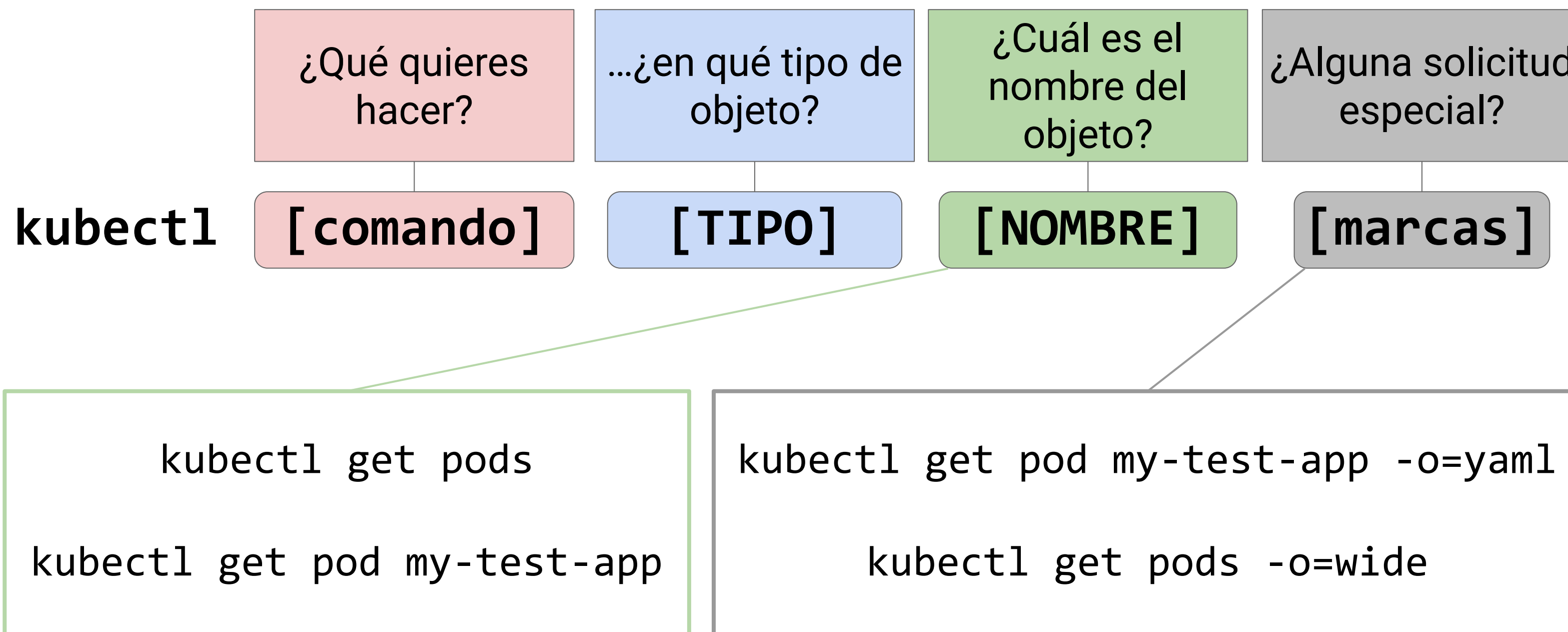
Conéctese a un clúster de Google Kubernetes Engine

```
$ 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

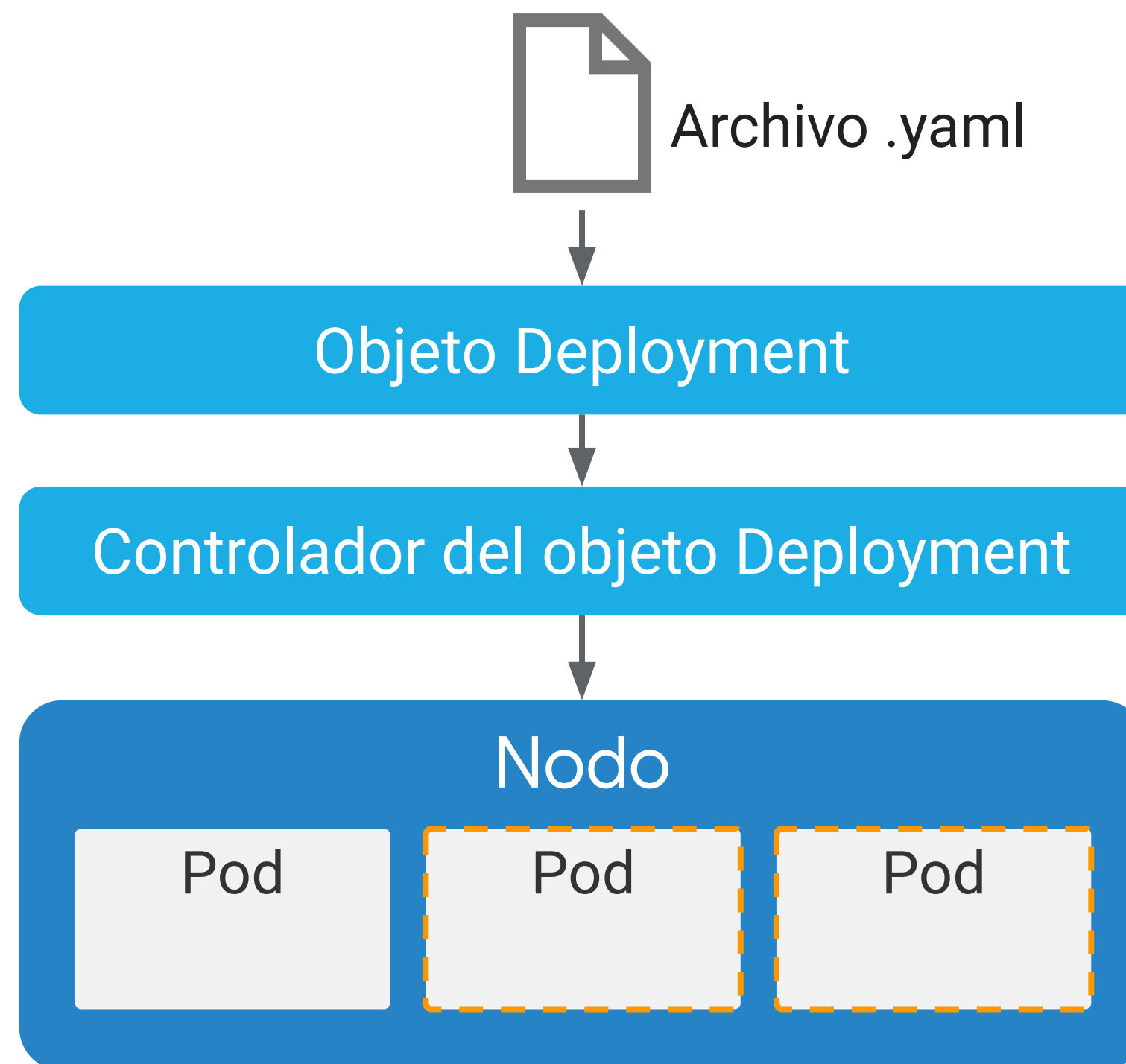


Escalamiento o ajuste de escala automático de los Pods



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
```

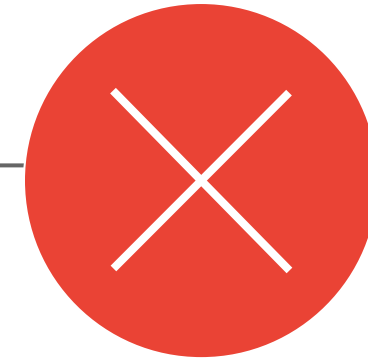
El objeto Deployment tiene tres estados distintos de ciclo de vida



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

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

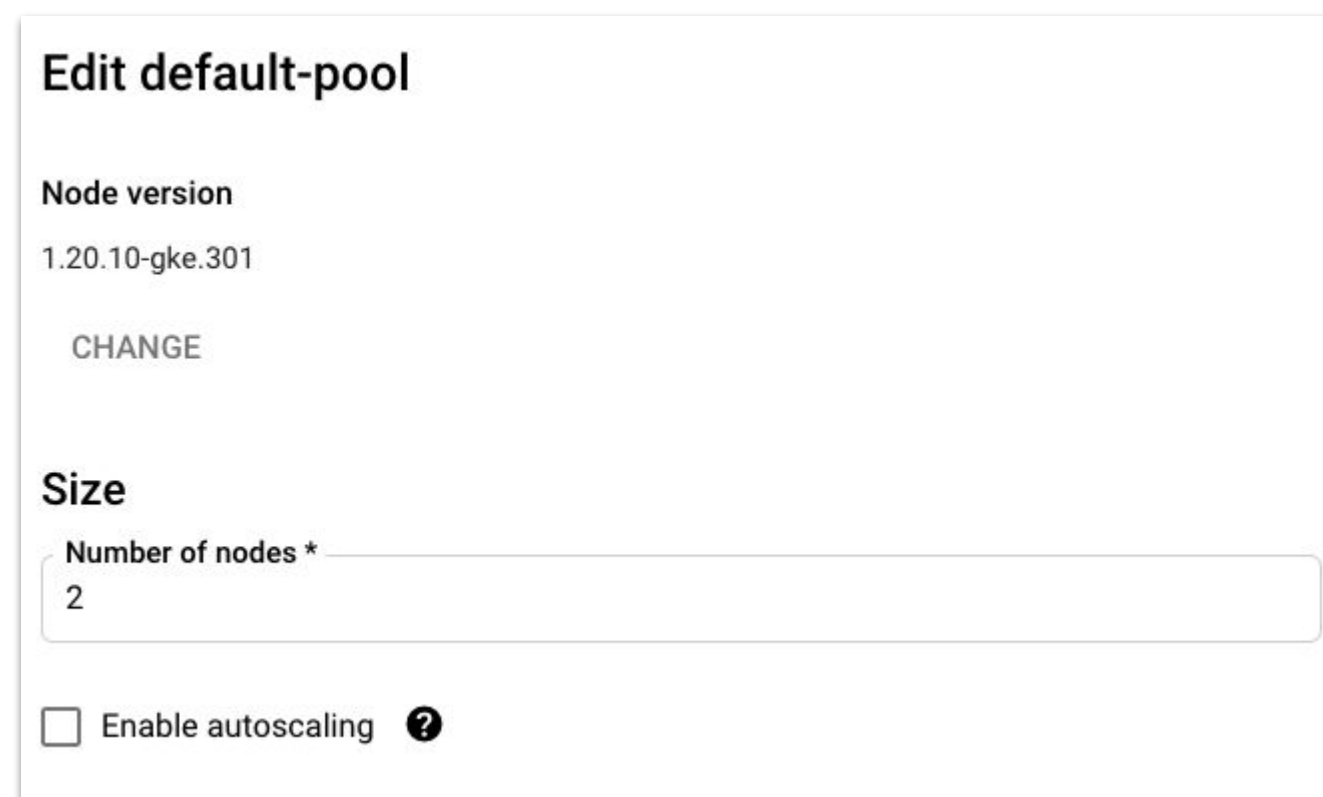
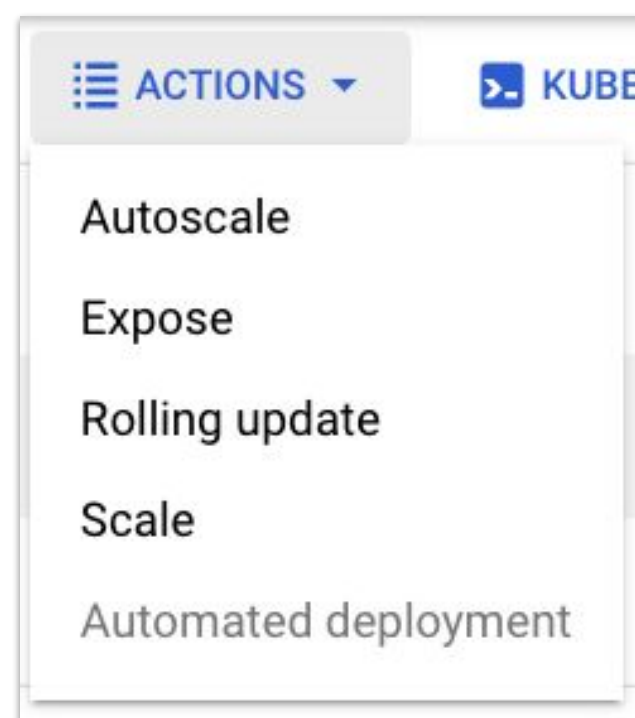
Containers

nginx

Google Cloud

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

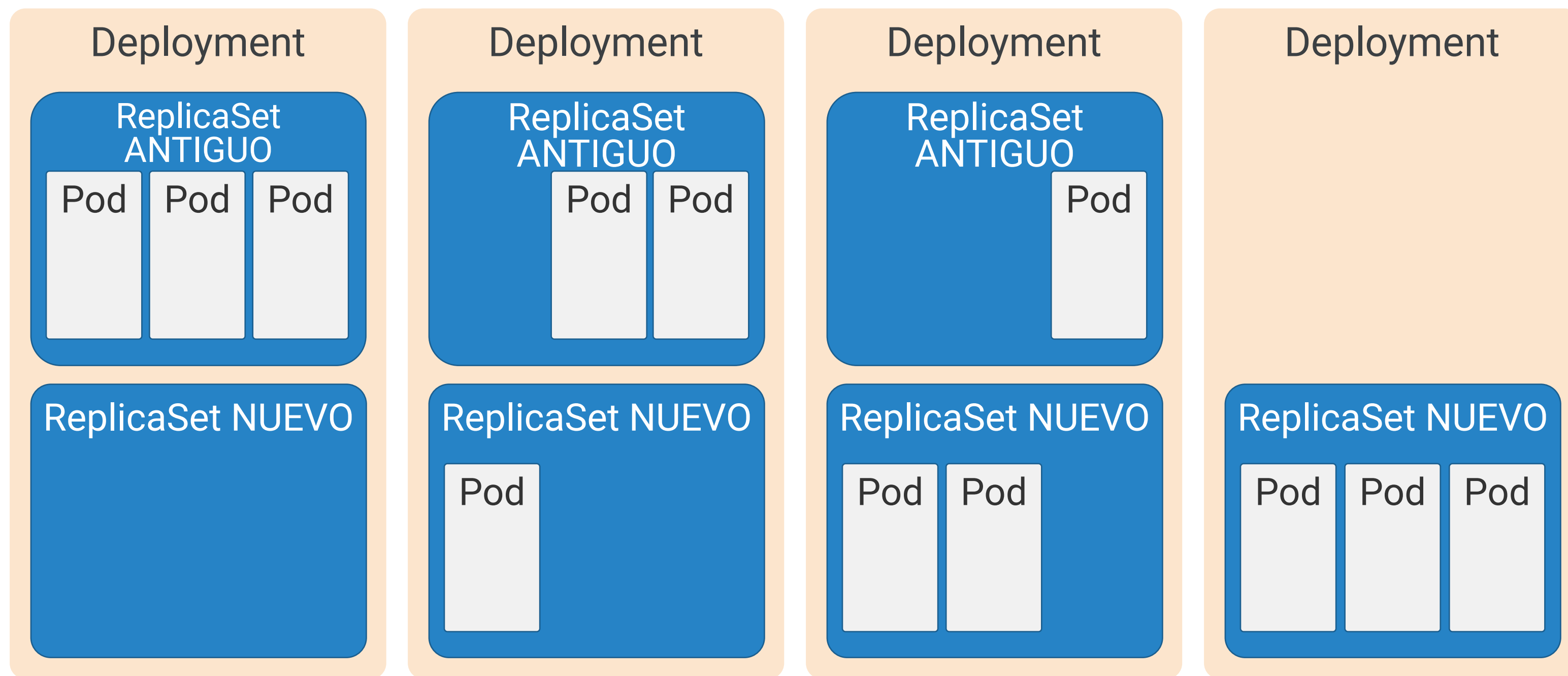
Image of nginx *

nginx:1.7.9

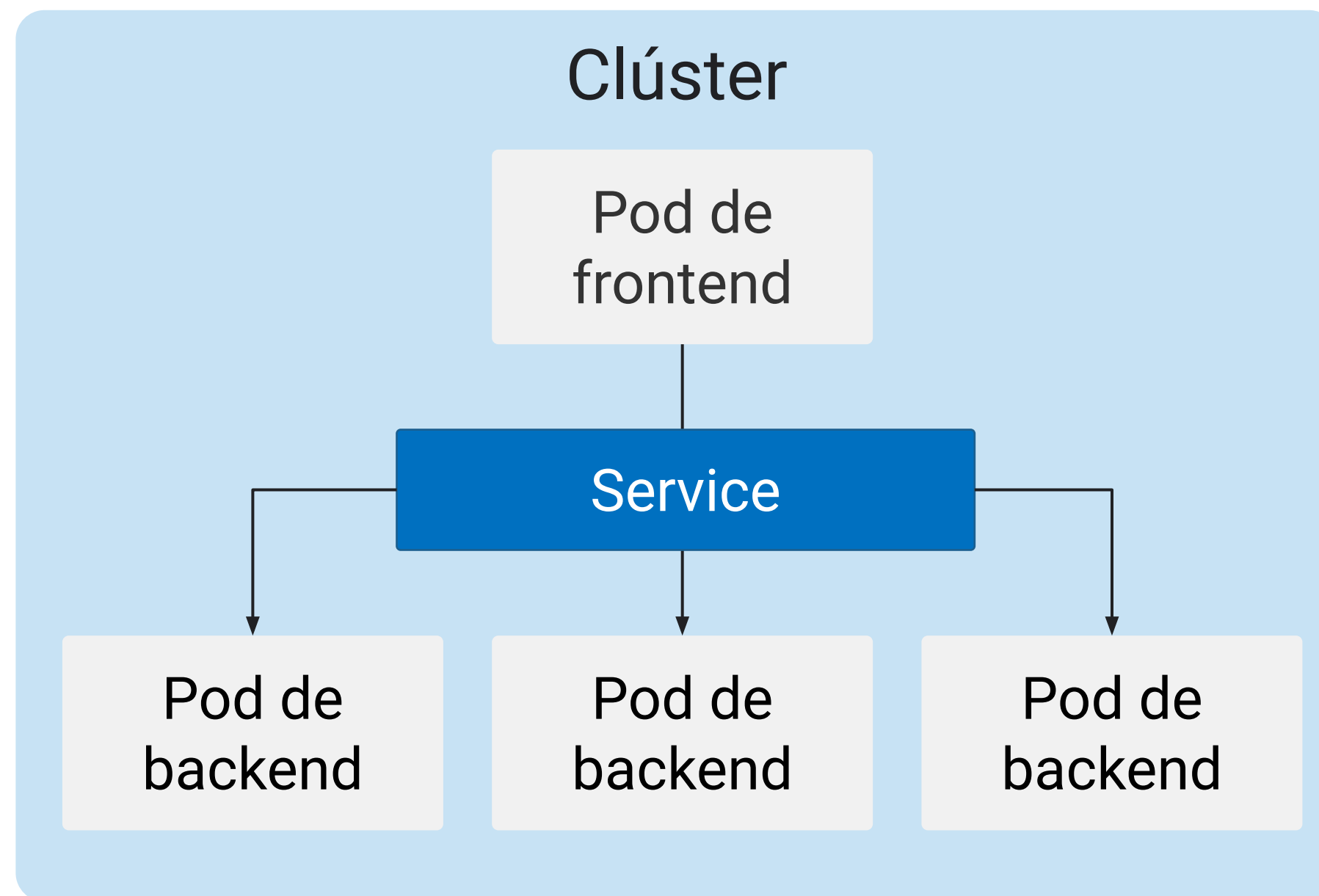
* Indicates required field

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

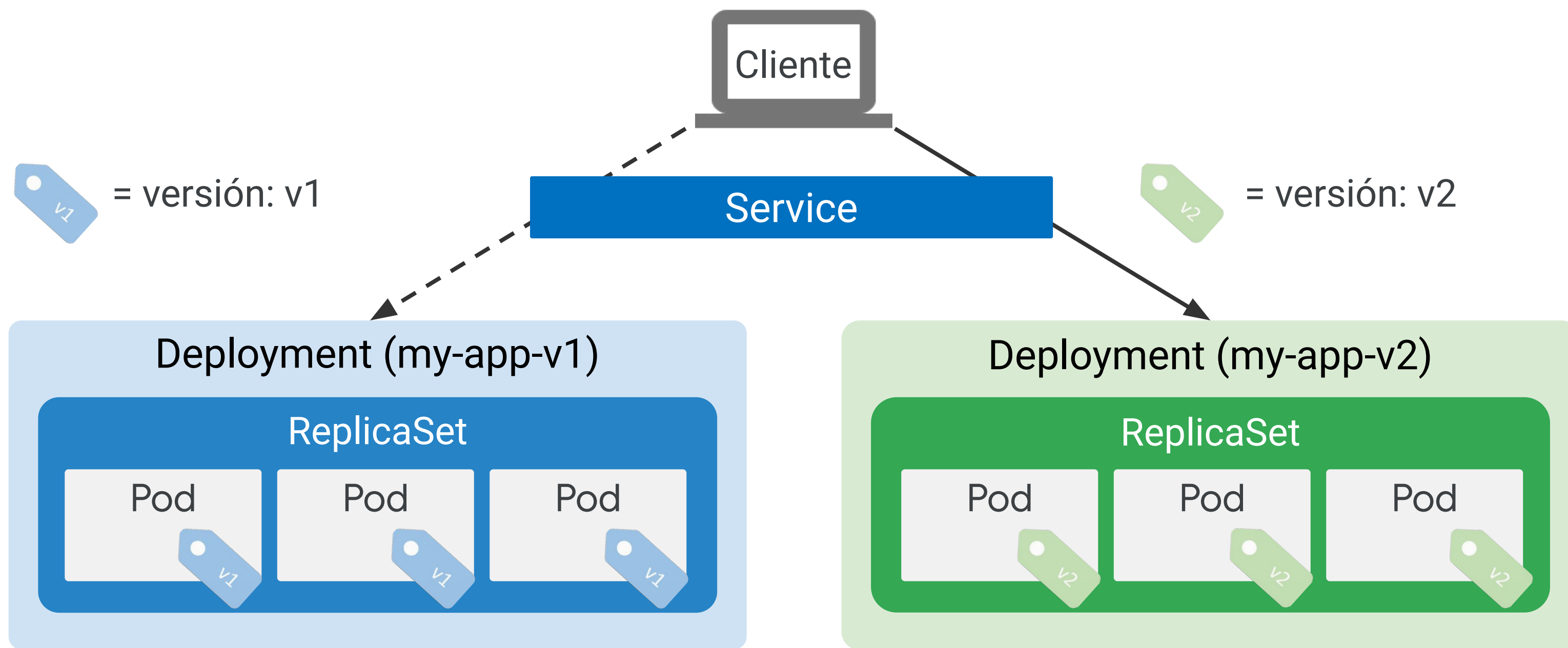
El proceso que conlleva la actualización de un objeto Deployment



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  
[...]
```

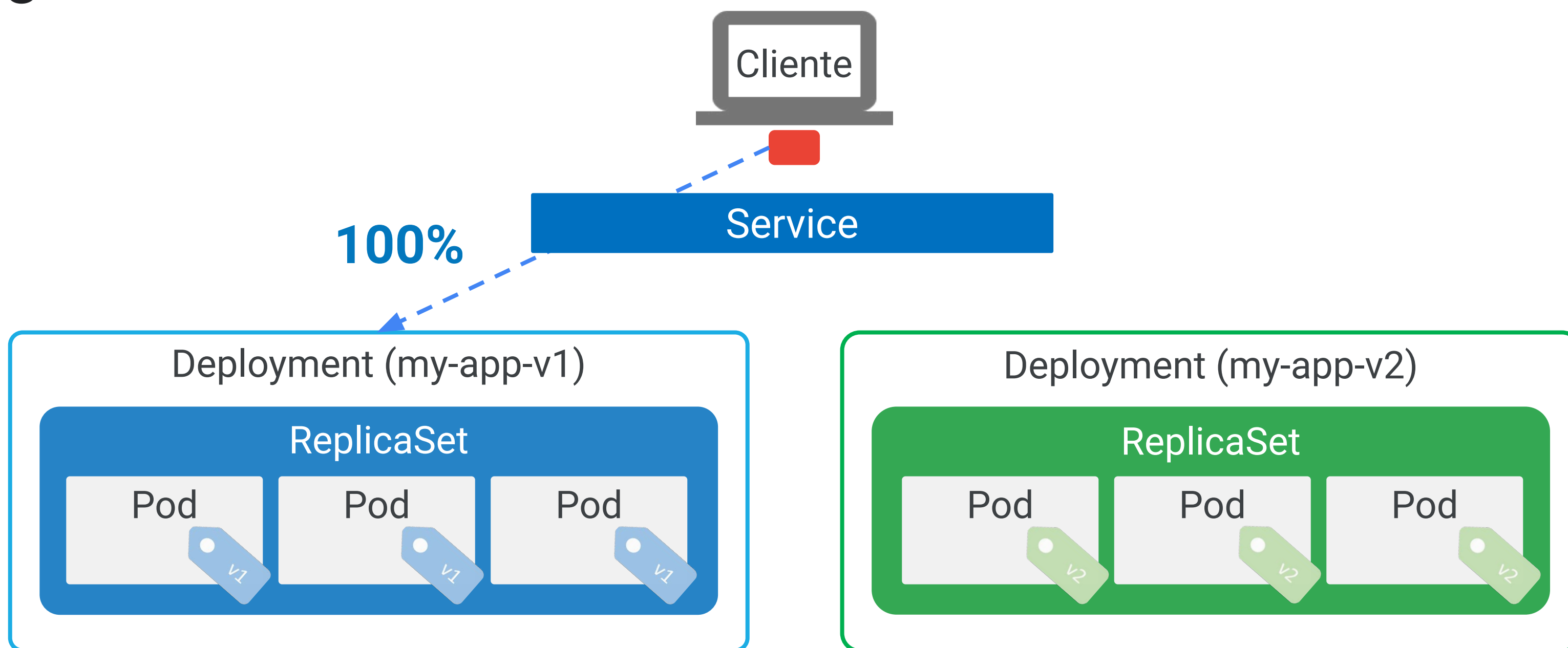
```
[...]  
kind: Service  
spec:  
  selector:  
    app: my-app  
    version: v2  
[...]
```



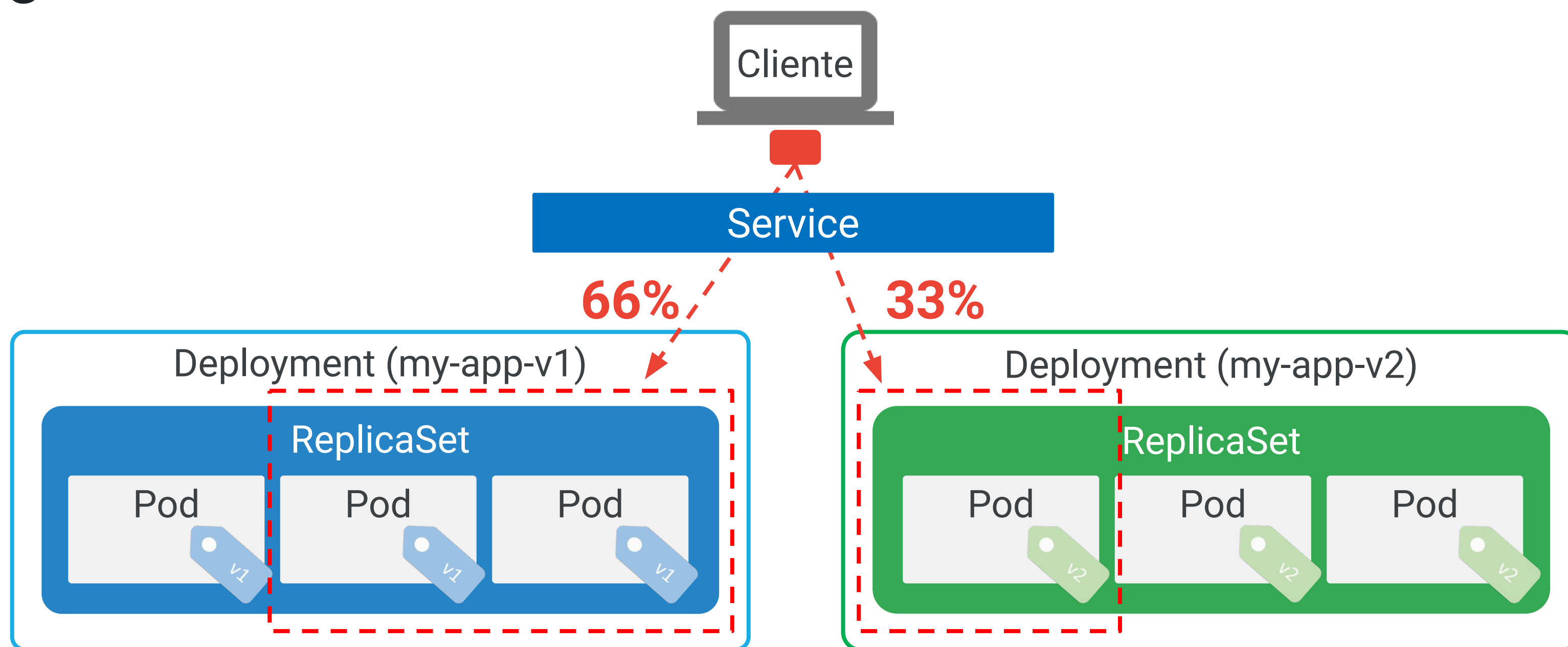
```
$ kubectl apply -f my-app-v2.yaml
```

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

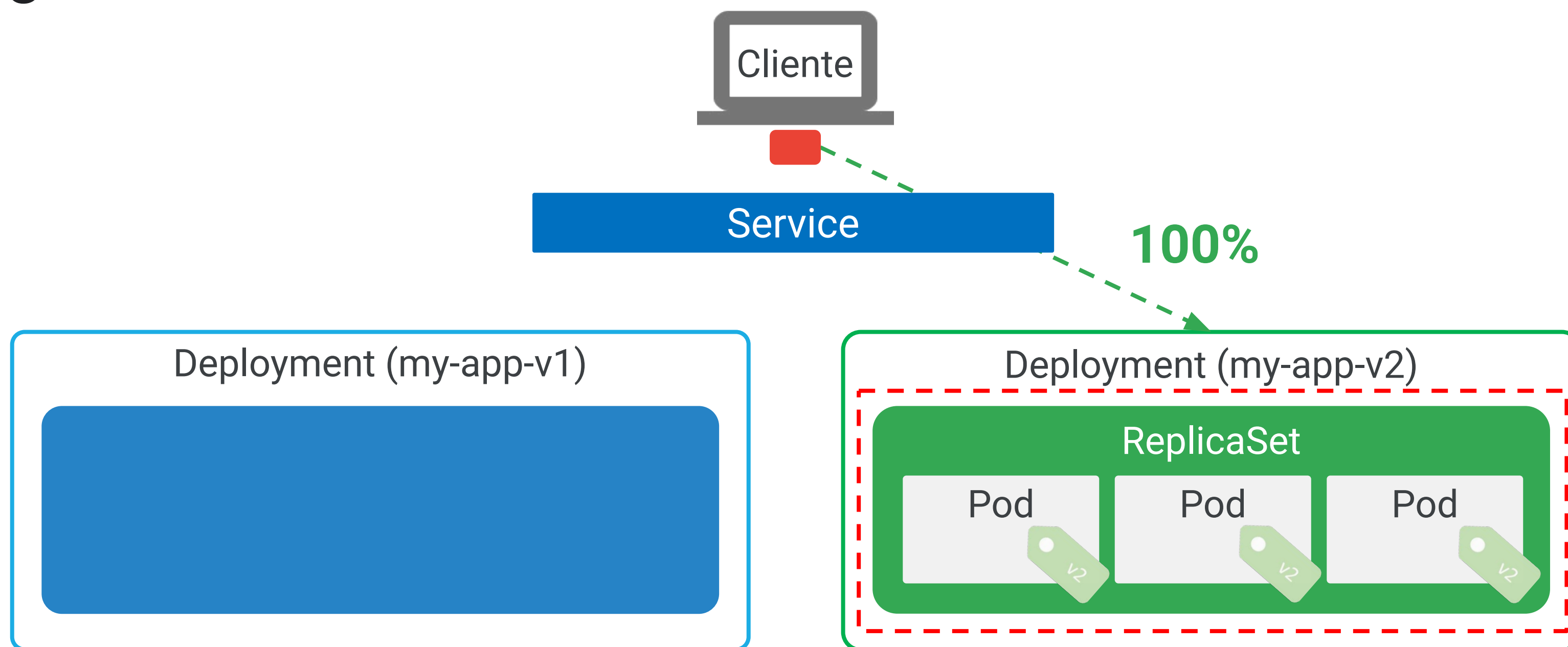
La implementación de versiones canary es una estrategia de actualización en la que el tráfico se cambia de forma gradual a la versión nueva



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

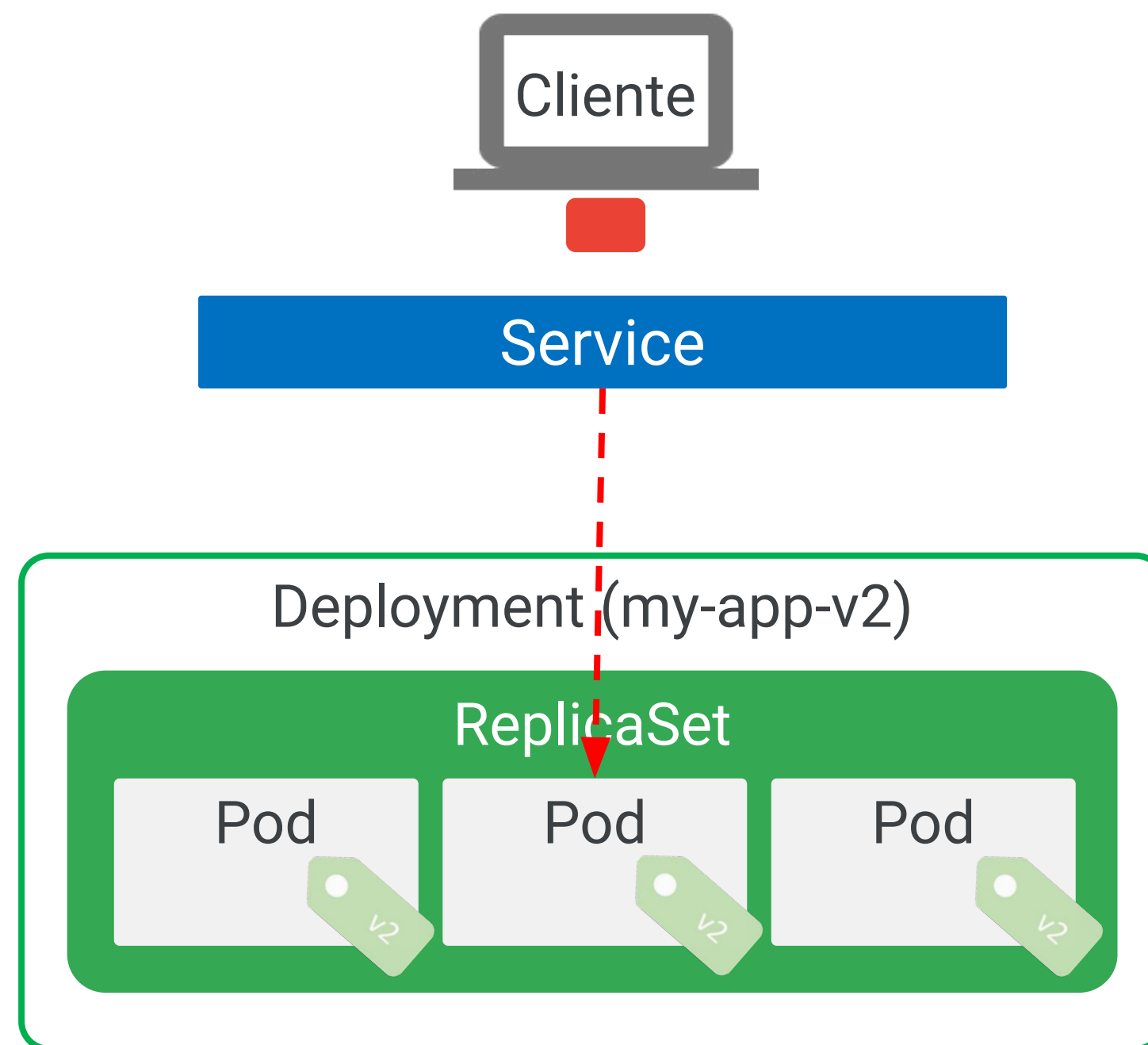
```
[...]  
kind: Service  
spec:  
  selector:  
    app: my-app  
[...]
```

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

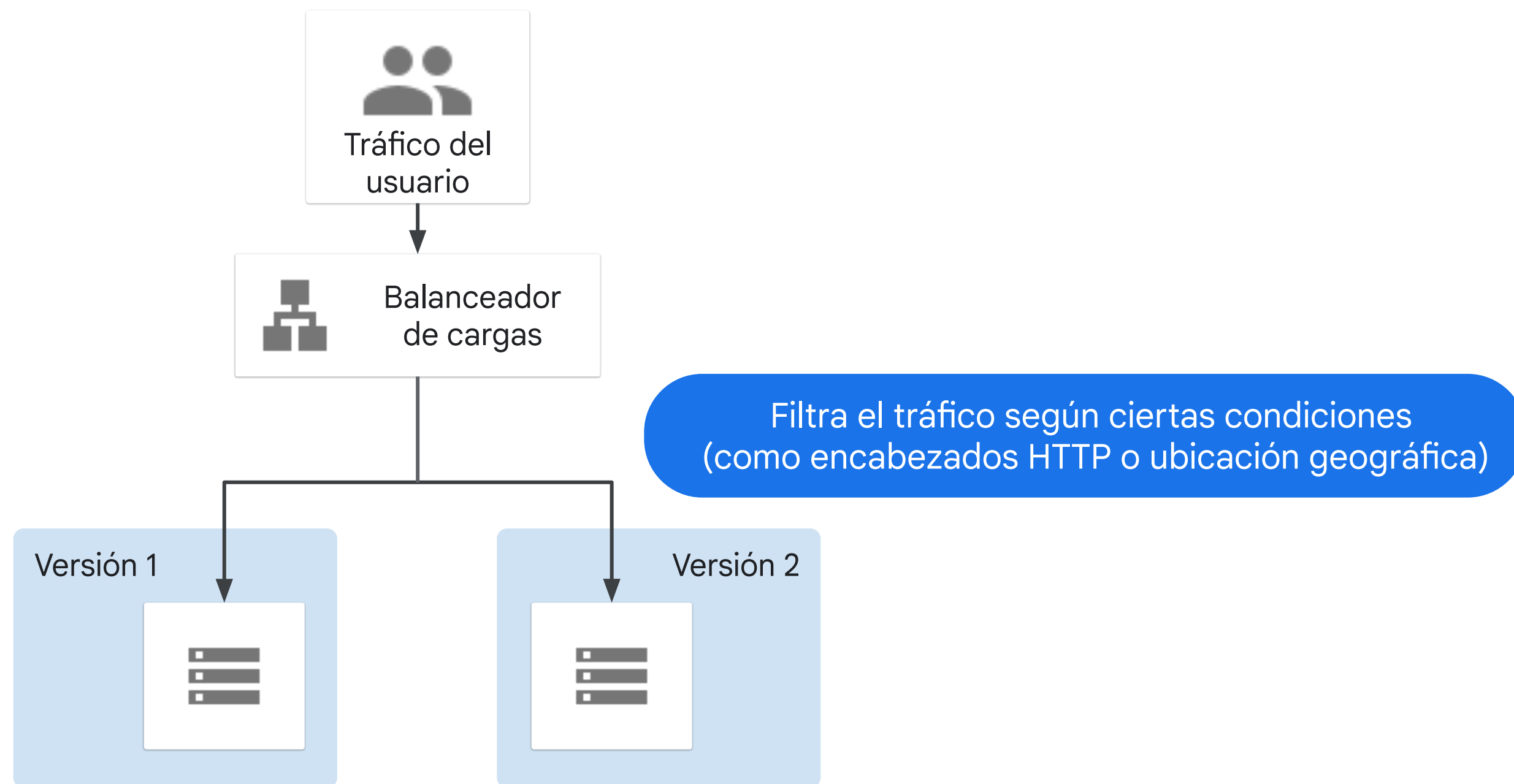
```
$ kubectl scale deploy/my-app-v2 --replicas=10
```

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

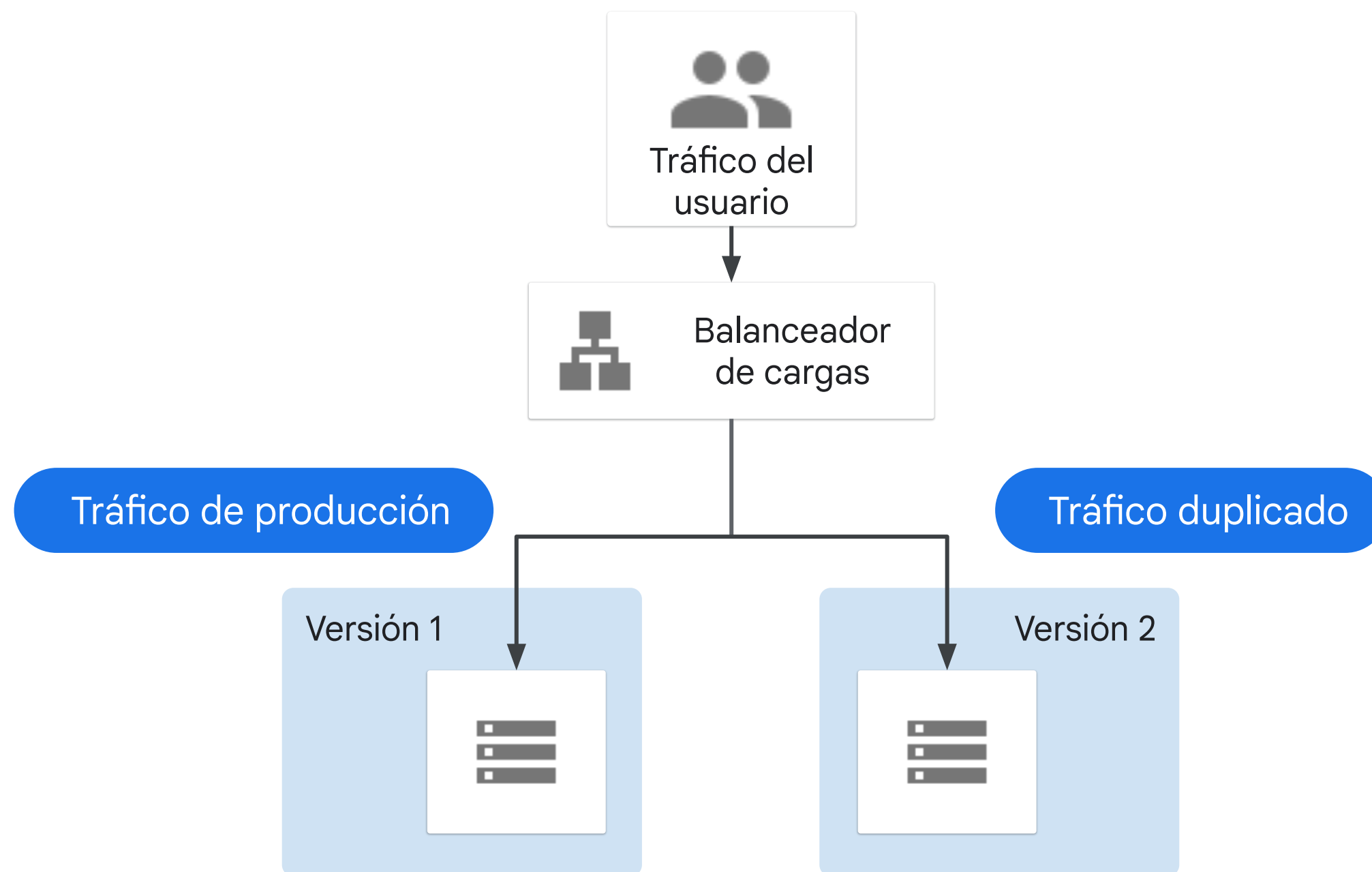

La afinidad de sesión garantiza que todas las solicitudes de clientes se envíen al mismo Pod



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



Temario

El comando de kubectl

Objetos Deployment

Lab: Cómo crear
implementaciones de
Google Kubernetes Engine

Redes de Pods

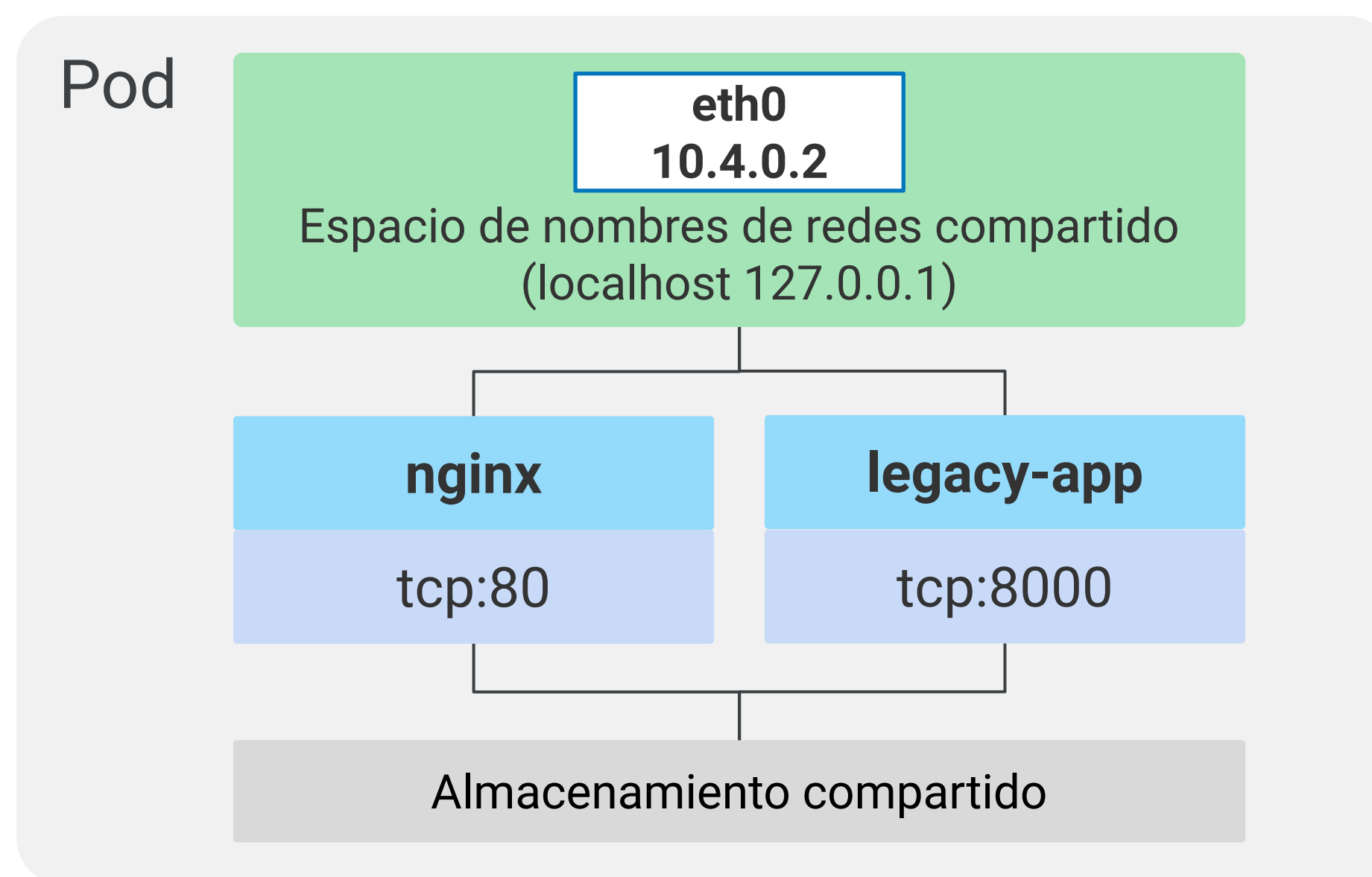
Volumes

Lab: Cómo configurar un
almacenamiento persistente para
Google Kubernetes Engine

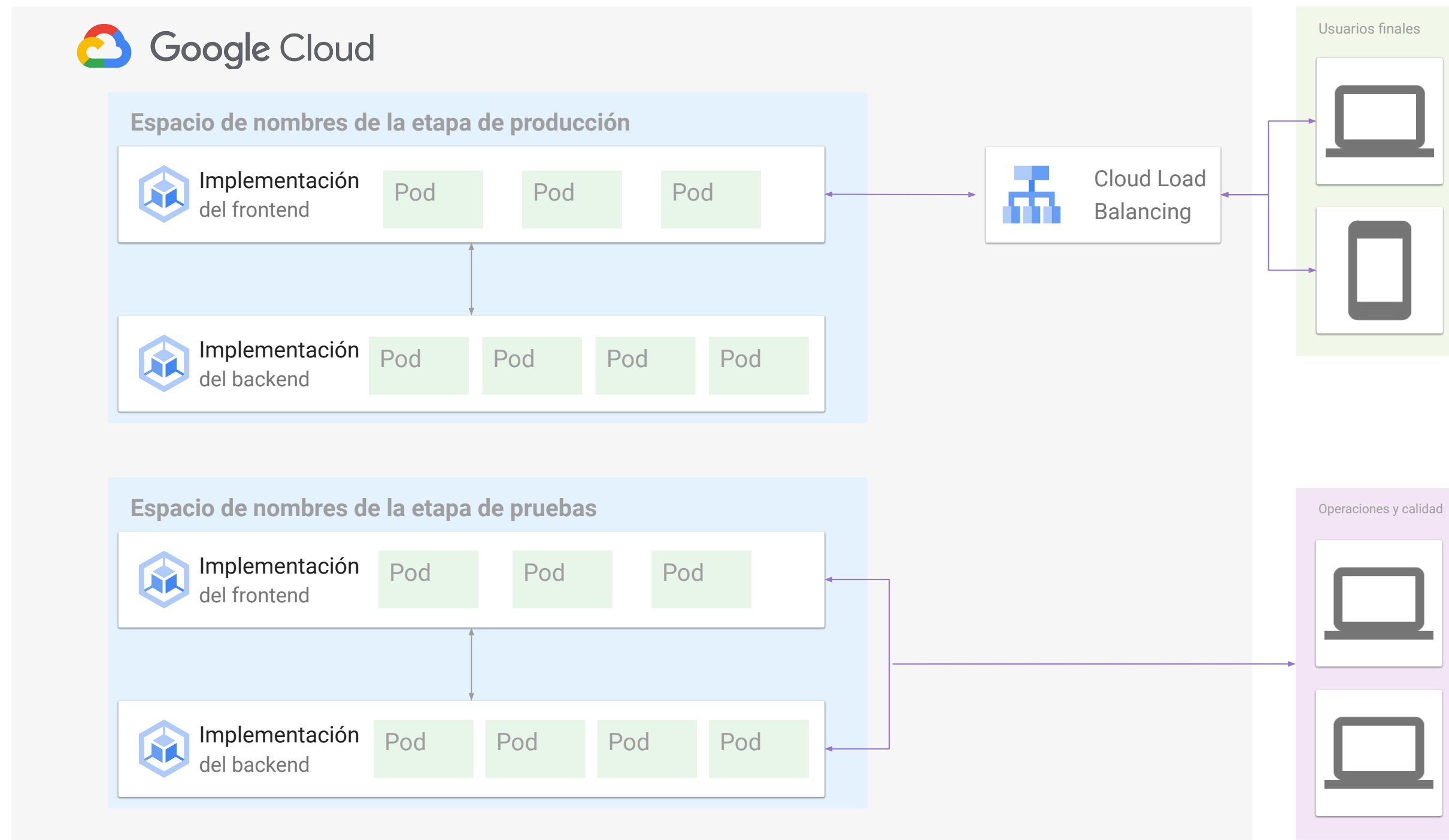
Cuestionario

Resumen

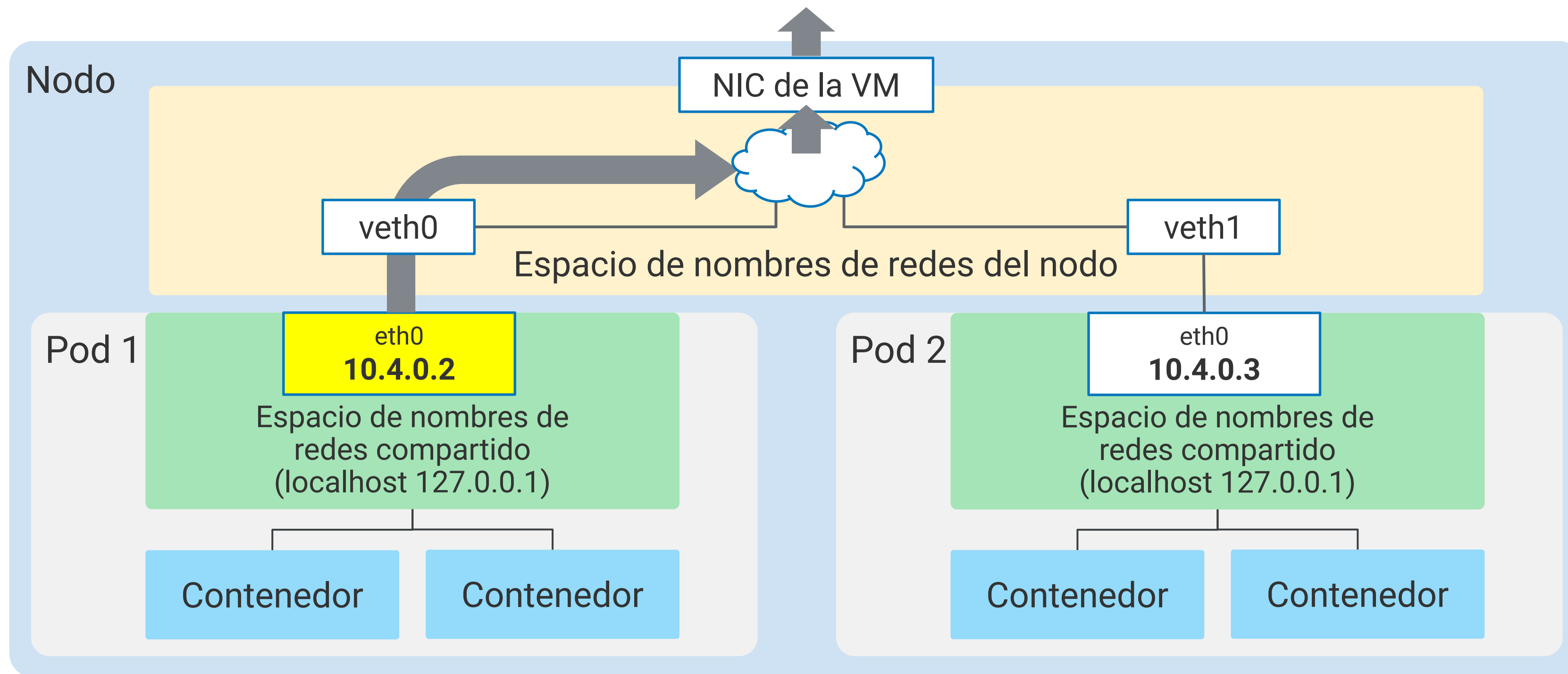
Un Pod es un grupo de contenedores con redes y almacenamiento compartidos



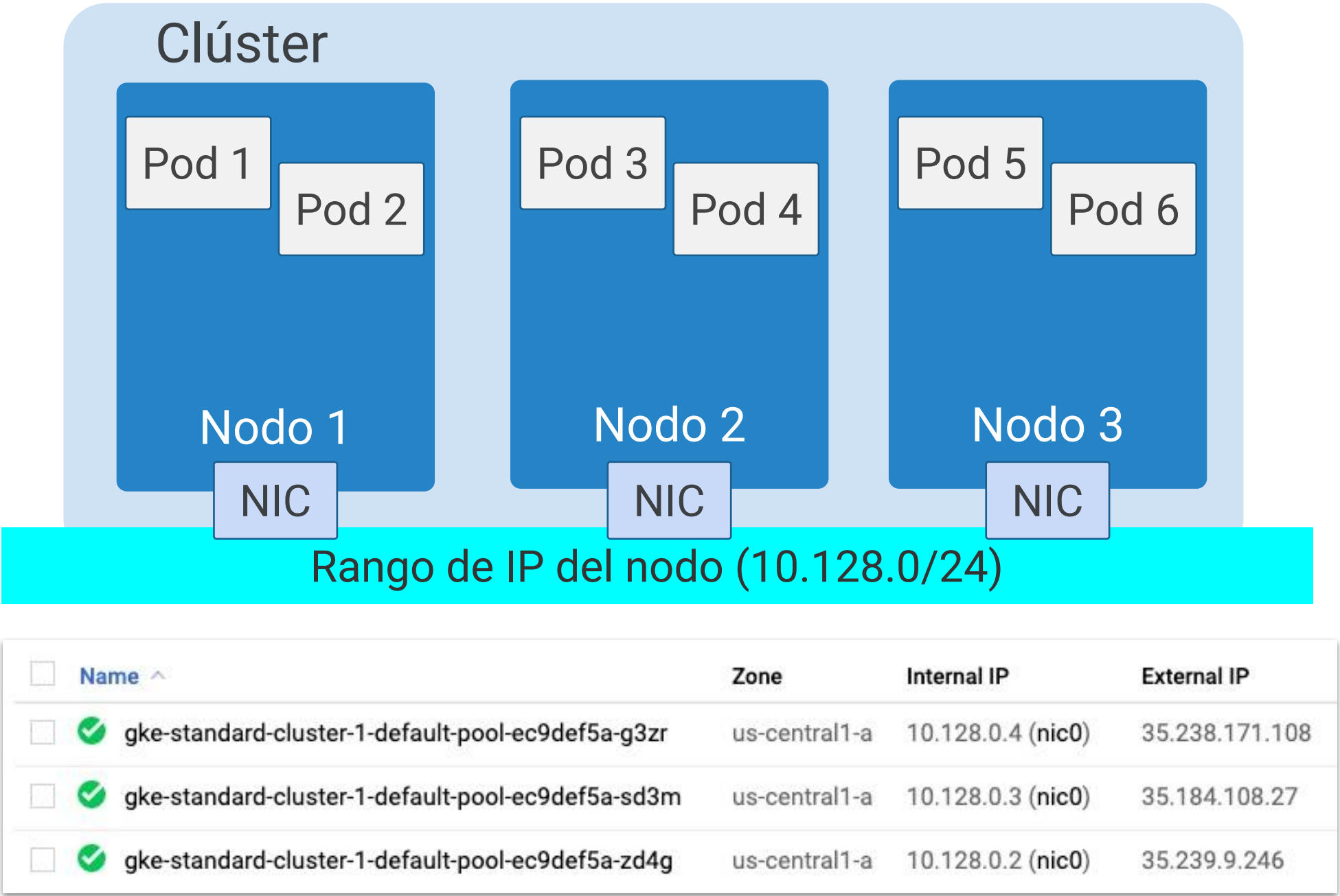
Su carga de trabajo no se ejecuta en un solo Pod



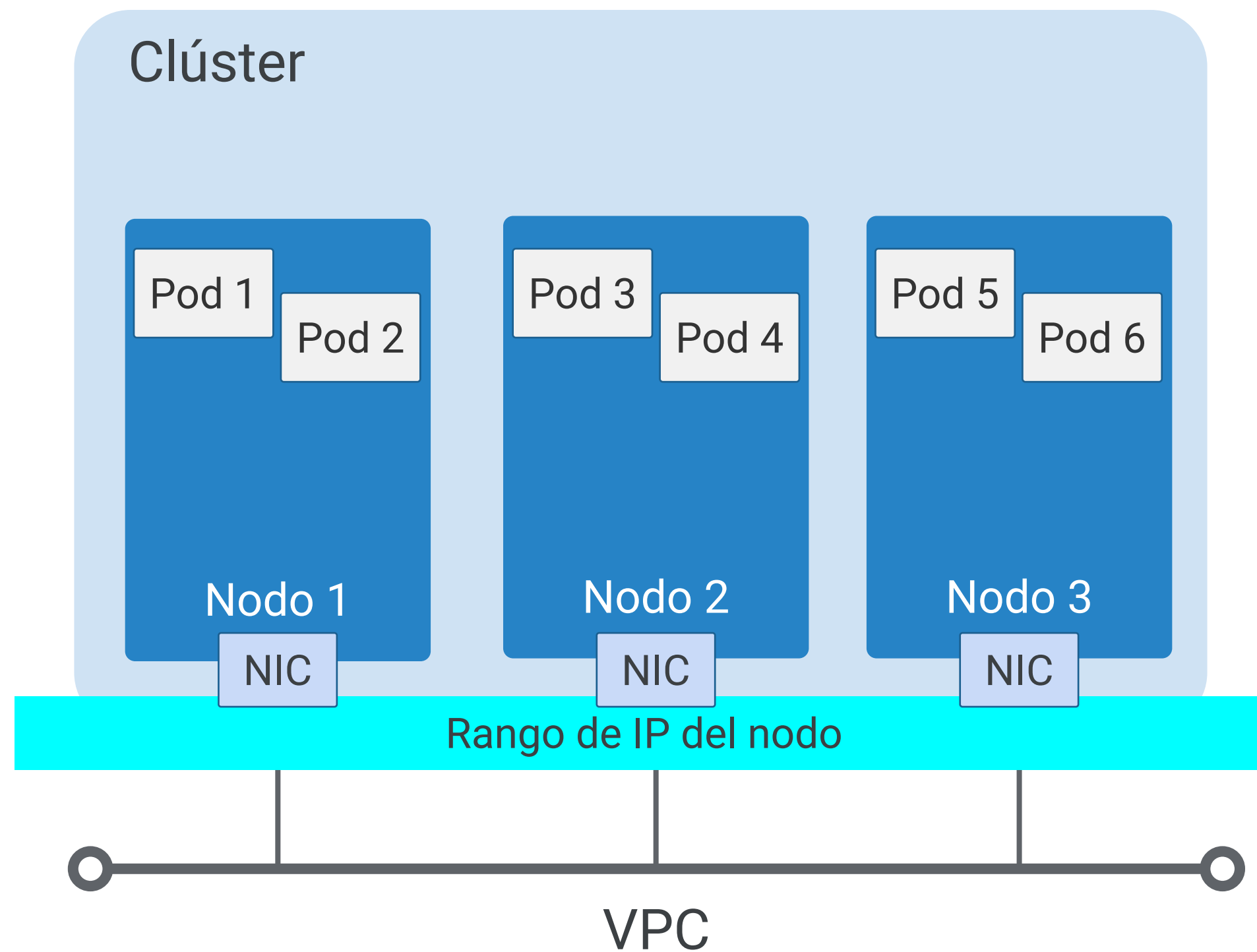
Comunicación entre Pods en el mismo nodo



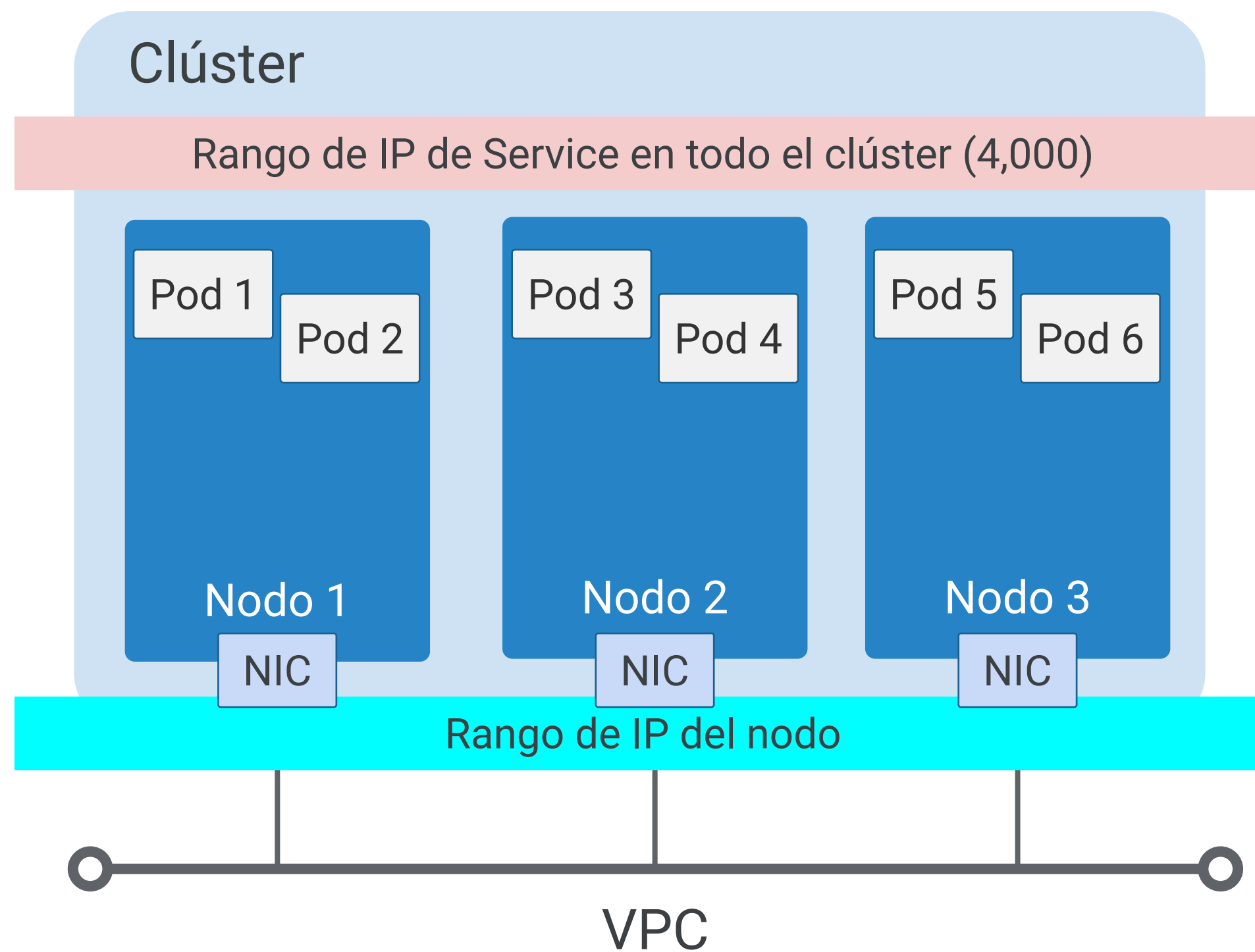
Los nodos obtienen las direcciones IP del Pod de los rangos de direcciones asignados a su nube privada virtual



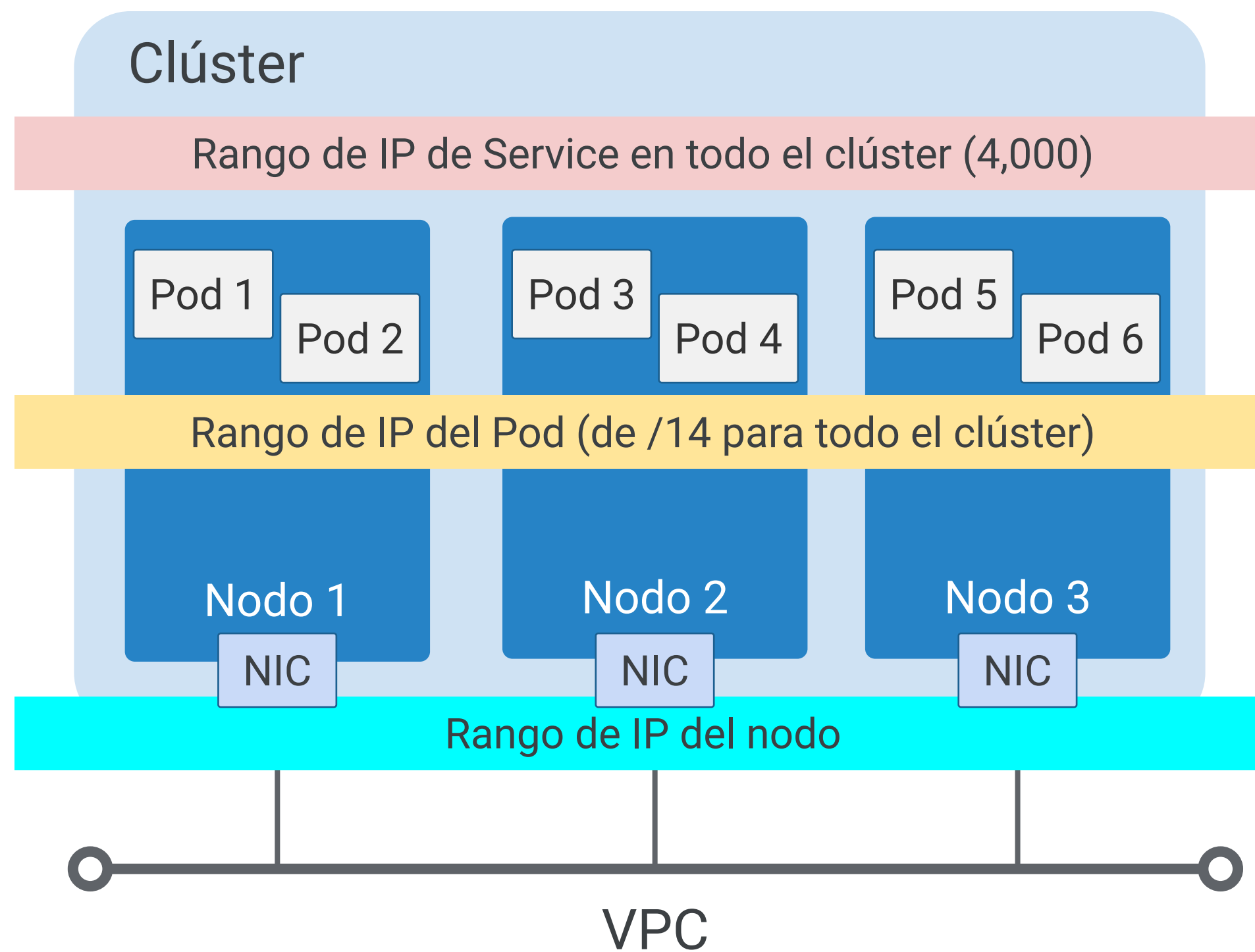
Cómo asignar direcciones a los Pods



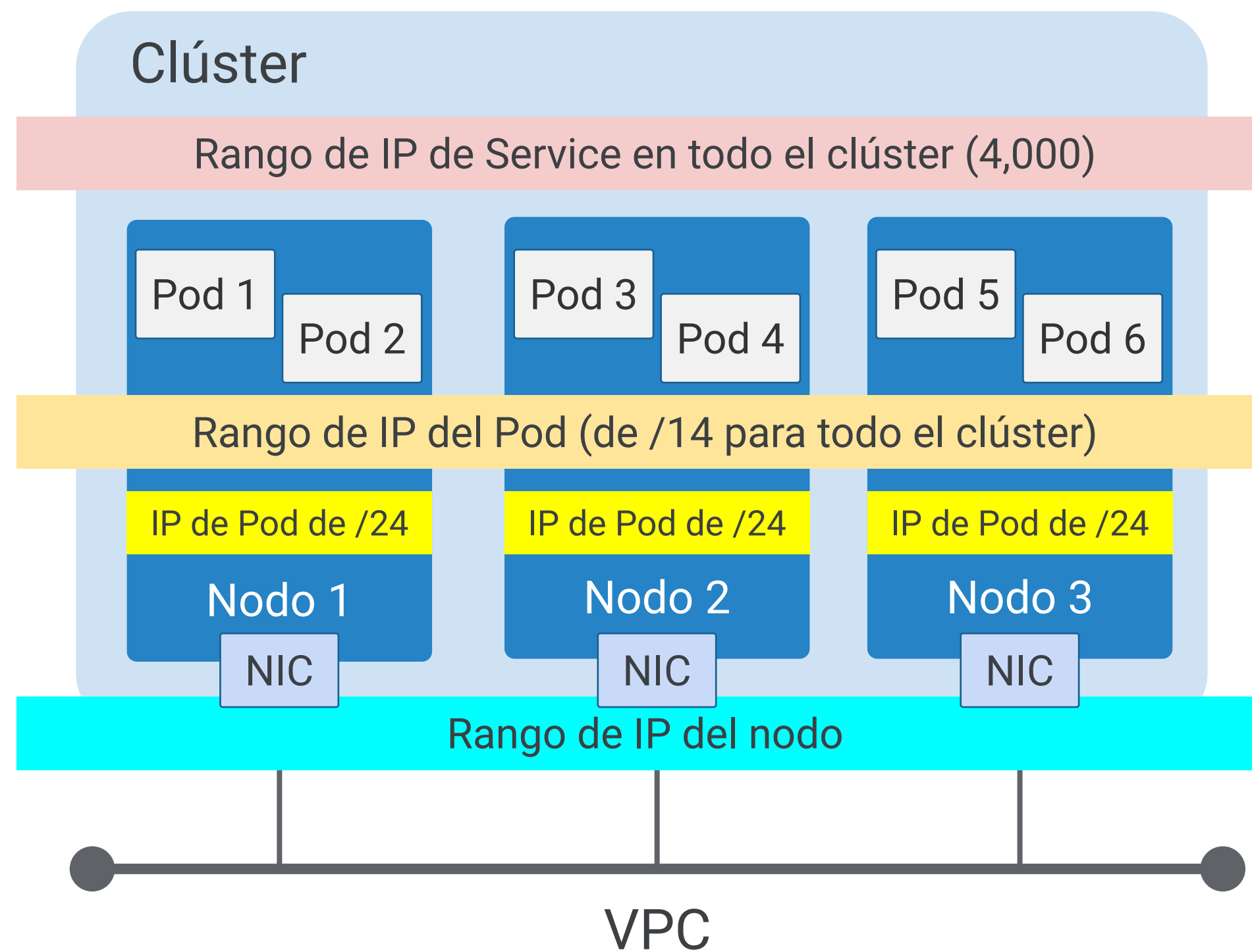
Cómo asignar direcciones a los Pods



Cómo asignar direcciones a los Pods



Cómo asignar direcciones a los Pods



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

Kubernetes ofrece opciones de abstracción de almacenamiento

Volumes

Son directorios a los que pueden acceder todos los contenedores de un Pod.

Algunos Volumes son efímeros.

Otros son persistentes.

PersistentVolumes

Permiten administrar almacenamiento durable en un clúster.

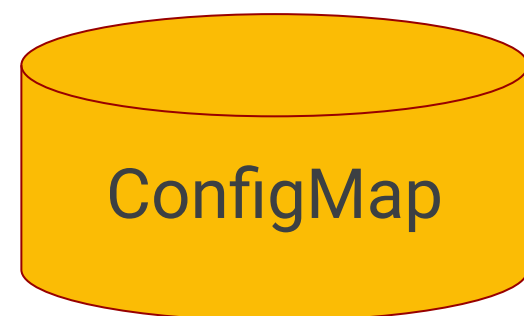
No dependen del ciclo de vida del Pod.

Se pueden aprovisionar de forma dinámica mediante PersistentVolumeClaims, o bien un administrador de clústeres los puede crear explícitamente.

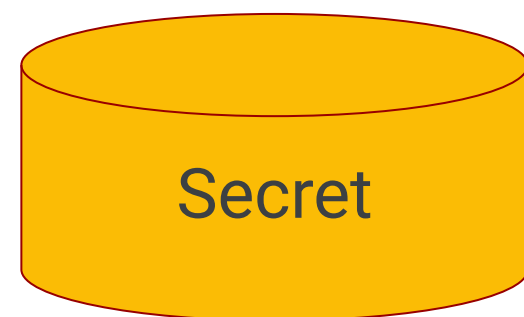
Explicación de los tipos de Volumes efímeros



Efímero: Comparte el ciclo de vida del Pod.



Se puede hacer referencia a un objeto en un Volume.

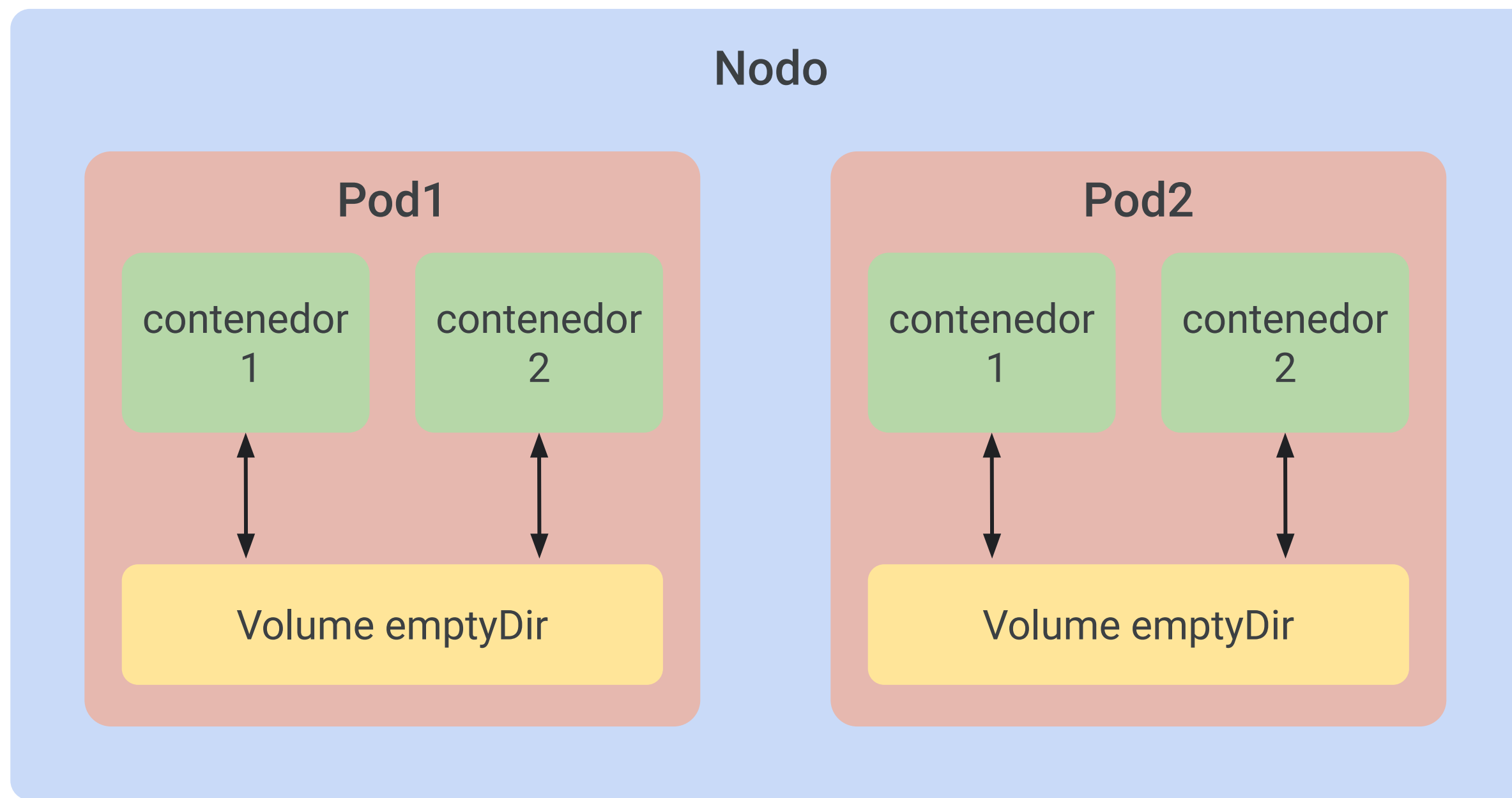


Almacena información sensible, como las contraseñas.



Permite que los datos sobre los Pods estén disponibles para los contenedores.

Cuando se asigna un Pod a un nodo, se crea un Volume emptyDir



Cómo crear un Pod con un Volume emptyDir

```
apiVersion: v1
kind: Pod
metadata:
  name: web
spec:
  containers:
  - name: web
    image: nginx
    volumeMounts:
    - mountPath: /cache
      name: cache-volume
  volumes:
  - name: cache-volume
    emptyDir: {}
```

Beneficios de los PersistentVolumes

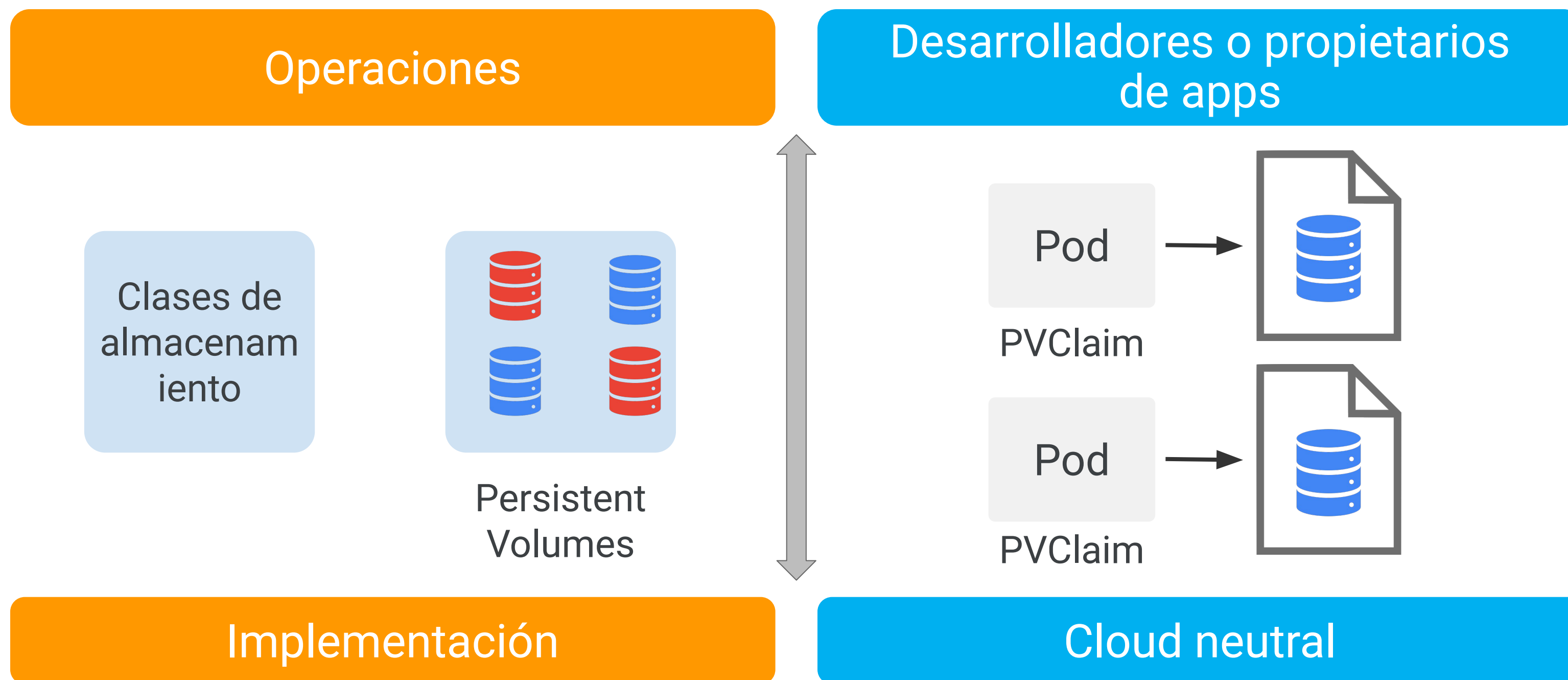
Abstraen el aprovisionamiento de almacenamiento del consumo de almacenamiento.

Promueven la arquitectura de microservicios.

Permiten a los administradores de clústeres aprovisionar y mantener el almacenamiento.

Los desarrolladores pueden reclamar el almacenamiento aprovisionado para el consumo de apps.

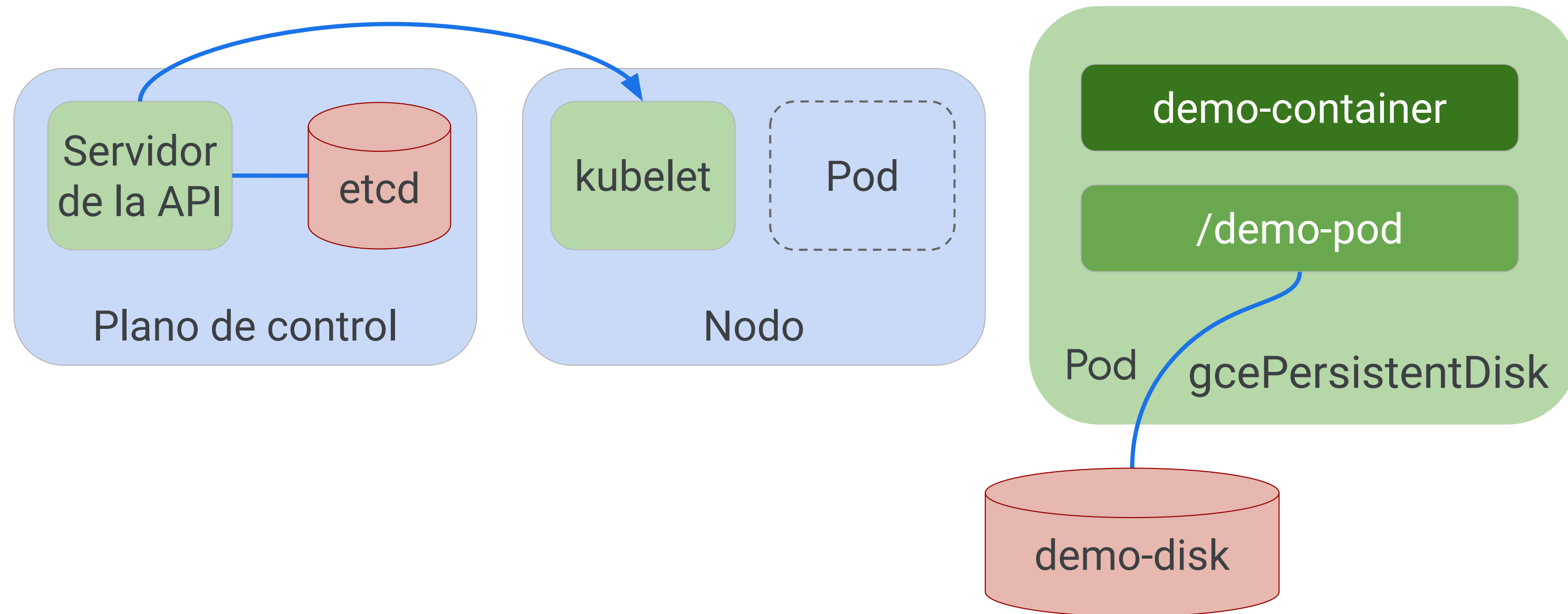
Las PersistentVolumeClaims y los PersistentVolumes separan el consumo de almacenamiento del aprovisionamiento



Cómo crear un disco persistente de Compute Engine con un comando de gcloud

```
$ gcloud compute disks create  
--size=100GB  
--zone=us-central1-a demo-disk
```

Cómo crear un disco persistente de Compute Engine



Configurar Volumes en Pods dificulta la portabilidad

```
Volumes:  
  pd-volume:  
    Type:          GCEPersistentDisk  
    PDName:        demo-disk  
    FSType:         ext4  
    Partition:     0
```

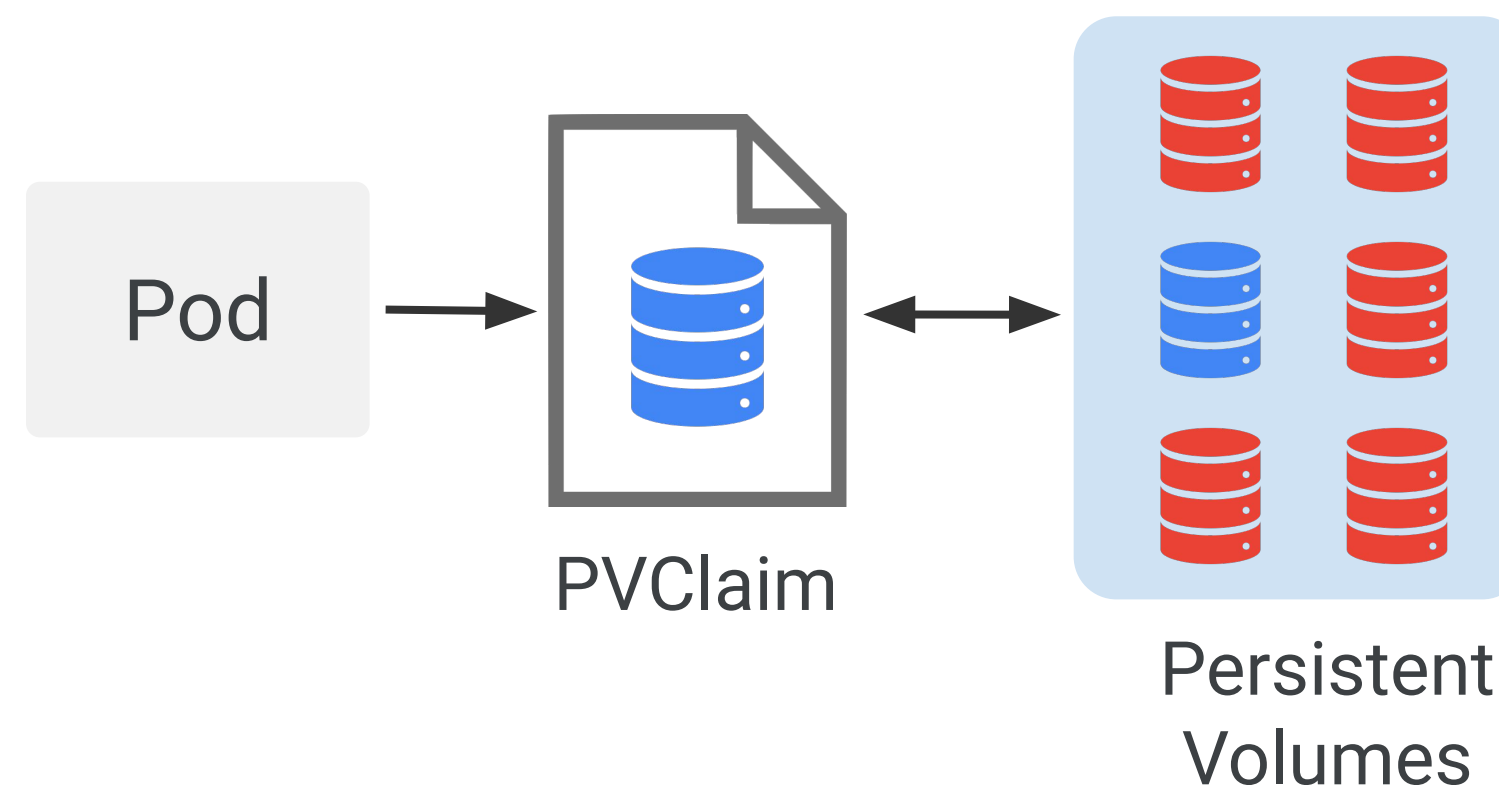
```
Volumes:  
  pd-volume:  
    Type:          vsphereVolume  
    PDName:        demo-disk  
    FSType:         ext4  
    Partition:     0
```

La abstracción PersistentVolumes tiene dos componentes

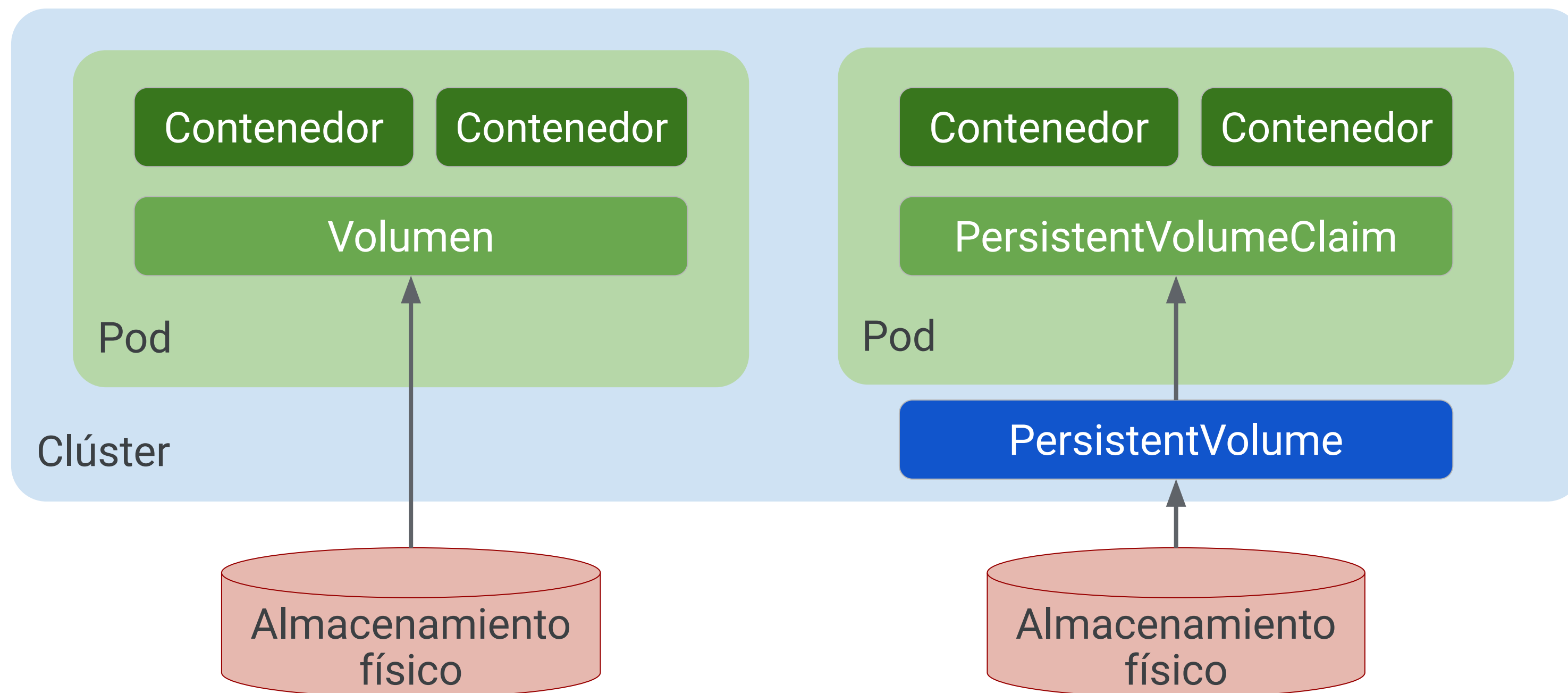
PersistentVolume (PV)

- No depende del ciclo de vida de un Pod.
- La administra Kubernetes.
- Se aprovisiona de forma manual o dinámica.
- GKE utiliza los discos persistentes como PersistentVolumes.

PersistentVolumeClaim (PVC)



Los PersistentVolumes deben reclamarse



Cómo crear un manifiesto de PersistentVolume

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pd-volume
spec:
  storageClassName: "standard"
  capacity:
    storage: 100G
  accessModes:
    - ReadWriteOnce:
  gcePersistentDisk:
    pdName: demo-disk
    fsType: ext4
```

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: standard
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-standard
  replication-type: none
```

PVC StorageClassName
debe coincidir con
PV StorageClassName

Crea una StorageClass nueva para usar un disco persistente SSD

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pd-volume
spec:
  storageClassName: "ssd"
  capacity:
    storage: 100G
  accessModes:
    - ReadWriteOnce:
  gcePersistentDisk:
    pdName: demo-disk
    fsType: ext4
```


```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ssd
provisioner: kubernetes.io/gce-pd
parameters:
  type: pd-ssd
```



El método moderno y más fácil de administrar es el uso de la abstracción PersistentVolumes


```
apiVersion: v1
kind: Pod
metadata:
  name: demo-pod
spec:
  containers:
  - name: demo-container
    image: gcr.io/hello-app:1.0
    volumeMounts:
    - mountPath: /demo-pod
      name: pd-volume
  volumes:
  - name: pd-volume
    PersistentVolumeClaim:
      claimName: pd-volume-claim
```


```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pd-volume-claim
spec:
  storageClassName: "standard"
  accessModes:
  - ReadWriteOnce:
  resources:
    requests:
      storage: 100G
```


Cómo visualizar la nueva clase de almacenamiento en la consola de Cloud


 Kubernetes Engine


 Clusters


 Workloads


 Services & Ingress


 Applications

 Configuration


 Storage


 Object Browser

 Migrate to containers

 Config Management

Storage

 REFRESH

 DELETE


ClusterNamespace

RESETSAVE

PERSISTENT VOLUME CLAIMS

STORAGE CLASSES

Storage classes are parameters for a class of storage for which persistent volumes are dynamically provisioned.

 Filter

Filter storage classes

<input type="checkbox"/>	Name ↑	Provisioner	Type	Zone	Cluster
<input type="checkbox"/>	premium-rwo	pd.csi.storage.gke.io	pd-ssd		standard-cluster-1
<input type="checkbox"/>	standard	kubernetes.io/gce-pd	pd-standard		standard-cluster-1
<input type="checkbox"/>	standard-rwo	pd.csi.storage.gke.io	pd-balanced		standard-cluster-1

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 configurar un
almacenamiento persistente para
Google Kubernetes Engine



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

Pregunta nº 1

Pregunta

¿Con qué componente del plano de control interactúa el comando de kubectl?

- A. etcd
- B. kubelet
- C. kube-apiserver
- D. La API de GKE

Pregunta nº 1

Respuesta

¿Con qué componente del plano de control interactúa el comando de kubectl?

- A. etcd
- B. kubelet
- C. kube-apiserver
- D. La API de GKE



Pregunta nº 2

Pregunta

Desea usar `kubectl` para configurar su clúster, pero primero debe configurarlo. ¿Dónde almacena su archivo de configuración el comando de `kubectl`?

- A. `kubectl` siempre solicita las credenciales al usuario antes de ejecutar comandos.
- B. `kubectl` usa los mismos tokens de autorización y de credenciales que las utilidades de `gcloud CLI`.
- C. Cuando es necesario, la información de configuración se almacena en variables de entorno en la shell actual.
- D. La información de configuración se almacena en el archivo `$HOME/.kube/config`.

Pregunta nº 2

Respuesta

Desea usar `kubectl` para configurar su clúster, pero primero debe configurarlo. ¿Dónde almacena su archivo de configuración el comando de `kubectl`?

- A. `kubectl` siempre solicita las credenciales al usuario antes de ejecutar comandos.
- B. `kubectl` usa los mismos tokens de autorización y de credenciales que las utilidades de `gcloud CLI`.
- C. Cuando es necesario, la información de configuración se almacena en variables de entorno en la shell actual.
- D. La información de configuración se almacena en el archivo `$HOME/.kube/config`.



Pregunta nº 3

Pregunta

Desea usar un comando `kubectl get` para identificar en qué nodo se está ejecutando cada Pod. ¿Qué comando debe ejecutar?

- A. `kubectl get nodes`
- B. `kubectl get nodes -o=yaml`
- C. `kubectl get pods`
- D. `kubectl get pods -o=wide`

Pregunta nº 3

Respuesta

Desea usar un comando `kubectl get` para identificar en qué nodo se está ejecutando cada Pod. ¿Qué comando debe ejecutar?

- A. `kubectl get nodes`
- B. `kubectl get nodes -o=yaml`
- C. `kubectl get pods`
- D. `kubectl get pods -o=wide`



Pregunta nº 4

Pregunta



¿Cuál es el propósito de Service? Elija todas las opciones verdaderas (en este caso, son 2).

- A. Permitirle seleccionar cómo se exponen los Pods.
- B. Permitirle restringir el consumo de recursos de los Pods.
- C. Proporcionar un extremo de red de balanceo de cargas para Pods.
- D. Proporcionar una manera de inspeccionar y diagnosticar el código que se ejecuta en un Pod.

Pregunta nº 4

Respuesta

¿Cuál es el propósito de Service? Elija todas las opciones verdaderas (en este caso, son 2).

- A. Permitirle seleccionar cómo se exponen los Pods. 
- B. Permitirle restringir el consumo de recursos de los Pods.
- C. Proporcionar un extremo de red de balanceo de cargas para Pods. 
- D. Proporcionar una manera de inspeccionar y diagnosticar el código que se ejecuta en un Pod.

Pregunta nº 5

Pregunta

Después de que se crea un objeto Deployment y de que se estén ejecutando sus Pods, ¿qué componente es responsable de garantizar que se inicie un Pod de reemplazo cuando un Pod falle o se expulse?

- A. DaemonSet
- B. Deployment
- C. ReplicaSet
- D. StatefulSet

Pregunta nº 5

Respuesta

Después de que se crea un objeto Deployment y de que se estén ejecutando sus Pods, ¿qué componente es responsable de garantizar que se inicie un Pod de reemplazo cuando un Pod falle o se expulse?

- A. DaemonSet
- B. Deployment
- C. ReplicaSet
- D. StatefulSet



Pregunta nº 6

Pregunta

¿Cuál es la relación entre los objetos Deployment y ReplicaSet?

- A. Un objeto Deployment configura un controlador de ReplicaSet para crear y mantener una versión específica de los Pods que especifica el objeto Deployment.
- B. Un objeto Deployment configura un controlador de ReplicaSet para crear y mantener todos los Pods que especifica el objeto Deployment, independientemente de su versión.
- C. Un ReplicaSet configura un controlador del objeto Deployment para crear y mantener una versión específica de los Pods que especifica el objeto Deployment.
- D. No hay relación; en el Kubernetes moderno, los controladores Replication normalmente se usan para mantener un conjunto de Pods en un estado de ejecución.

Pregunta nº 6

Respuesta

¿Cuál es la relación entre los objetos Deployment y ReplicaSet?

- A. Un objeto Deployment configura un controlador de ReplicaSet para crear y mantener una versión específica de los Pods que especifica el objeto Deployment.
- B. Un objeto Deployment configura un controlador de ReplicaSet para crear y mantener todos los Pods que especifica el objeto Deployment, independientemente de su versión.
- C. Un ReplicaSet configura un controlador del objeto Deployment para crear y mantener una versión específica de los Pods que especifica el objeto Deployment.
- D. No hay relación; en el Kubernetes moderno, los controladores Replication normalmente se usan para mantener un conjunto de Pods en un estado de ejecución.



Pregunta nº 7

Pregunta

¿Qué tipo de aplicación es adecuada para usar con un objeto Deployment?

- A. Por lotes
- B. Con estado
- C. Sin estado
- D. Escrita en Go

Pregunta nº 7

Respuesta

¿Qué tipo de aplicación es adecuada para usar con un objeto Deployment?

- A. Por lotes
- B. Con estado
- C. Sin estado
- D. Escrita en Go



Pregunta nº 8

Pregunta

Realizó varias modificaciones en su implementación y aplicó los cambios. ¿Qué comando debe usar para revertir el entorno a la implementación identificada en el historial de implementaciones como la revisión 2?

- A. Debe ejecutar `"kubectl apply -f DEPLOYMENT_FILE --to-revision=2"`.
- B. Debe ejecutar `"kubectl rollout undo deployment"` dos veces.
- C. Debe ejecutar `"kubectl rollout undo deployment --to-revision=2"`.
- D. Debes seleccionar la revisión deseada de la lista del historial de revisión en la consola de Cloud.

Pregunta nº 8

Respuesta

Realizó varias modificaciones en su implementación y aplicó los cambios. ¿Qué comando debe usar para revertir el entorno a la implementación identificada en el historial de implementaciones como la revisión 2?

- A. Debe ejecutar `kubectl apply -f DEPLOYMENT_FILE --to-revision=2`.
- B. Debe ejecutar `kubectl rollout undo deployment` dos veces.
- C. Debe ejecutar `kubectl rollout undo deployment --to-revision=2`.
- D. Debes seleccionar la revisión deseada de la lista del historial de revisión en la consola de Cloud.



Pregunta nº 9

Pregunta

Estás resolviendo una variedad de problemas con un objeto Deployment y debes realizar una gran cantidad de cambios. ¿Qué comando puede ejecutar para agrupar estos cambios en un solo lanzamiento y, así, evitar aplicar una gran cantidad de lanzamientos?

- A. `kubectl delete deployment`
- B. `kubectl rollout pause deployment`
- C. `kubectl rollout resume deployment`
- D. `kubectl stop deployment`

Pregunta nº 9

Respuesta

Estás resolviendo una variedad de problemas con un objeto Deployment y debes realizar una gran cantidad de cambios. ¿Qué comando puede ejecutar para agrupar estos cambios en un solo lanzamiento y, así, evitar aplicar una gran cantidad de lanzamientos?

- A. `kubectl delete deployment`
- B. `kubectl rollout pause deployment`
- C. `kubectl rollout resume deployment`
- D. `kubectl stop deployment`



Pregunta nº 10

Pregunta

En GKE, ¿cuál es la fuente de las direcciones IP para Pods?

- A. Rangos de direcciones asignados a su nube privada virtual
- B. Direcciones de red arbitrarias por clúster
- C. Direcciones de red de bucle invertido

Pregunta nº 10

Respuesta

En GKE, ¿cuál es la fuente de las direcciones IP para Pods?

- A. Rangos de direcciones asignados a su nube privada virtual
- B. Direcciones de red arbitrarias por clúster
- C. Direcciones de red de bucle invertido



Pregunta nº 11

Pregunta

Se reprogramó su Pod y ya no es posible acceder a la dirección IP que se le asignó cuando se programó originalmente. ¿Cuál puede ser el motivo?

- A. Un firewall bloquea la nueva dirección IP del Pod.
- B. El nuevo Pod recibió una dirección IP diferente.
- C. Un firewall bloquea la antigua dirección IP del Pod.
- D. Se agotó el rango de IP del Pod para el clúster.

Pregunta nº 11

Respuesta

Se reprogramó su Pod y ya no es posible acceder a la dirección IP que se le asignó cuando se programó originalmente. ¿Cuál puede ser el motivo?

- A. Un firewall bloquea la nueva dirección IP del Pod.
- B. El nuevo Pod recibió una dirección IP diferente.**
- C. Un firewall bloquea la antigua dirección IP del Pod.
- D. Se agotó el rango de IP del Pod para el clúster.



Pregunta nº 12

Pregunta

¿Qué sucede si falla un Pod mientras está usando un volumen persistente?

- A. Se borran los volúmenes y se pierde su contenido.
- B. Los volúmenes se desactivan en el Pod que falla y se borra su contenido.
- C. Los volúmenes se desactivan en el Pod que falla y su contenido se revierte a lo que tenían antes de que el Pod se adjuntara a ellos.
- D. Los volúmenes se desactivan en el Pod que falla y continúan existiendo con su último contenido.

Pregunta nº 12

Respuesta

¿Qué sucede si falla un Pod mientras está usando un volumen persistente?

- A. Se borran los volúmenes y se pierde su contenido.
- B. Los volúmenes se desactivan en el Pod que falla y se borra su contenido.
- C. Los volúmenes se desactivan en el Pod que falla y su contenido se revierte a lo que tenían antes de que el Pod se adjuntara a ellos.
- D. Los volúmenes se desactivan en el Pod que falla y continúan existiendo con su último contenido.



Pregunta nº 13

Pregunta

¿Cómo puede un Pod solicitar almacenamiento persistente sin especificar los detalles de cómo se implementará dicho almacenamiento?

- A. Con `gcePersistentDisk`
- B. Con `PersistentVolume`
- C. Con `PersistentVolumeClaim`
- D. Con `emptyDir`

Pregunta nº 13

Respuesta

¿Cómo puede un Pod solicitar almacenamiento persistente sin especificar los detalles de cómo se implementará dicho almacenamiento?

- A. Con `gcePersistentDisk`
- B. Con `PersistentVolume`
- C. Con `PersistentVolumeClaim`
- D. Con `emptyDir`



Pregunta nº 14

Pregunta

El propietario de una aplicación creó un manifiesto de Pod mediante una `PersistentVolumeClaim` con un valor estándar de `StorageClassName`. ¿Qué tipo de almacenamiento se usa para este volumen en un clúster de GKE?

- A. Google Persistent Disk
- B. Volumen local en el nodo
- C. Con copia de seguridad de una memoria
- D. Almacenamiento de NFS

Pregunta nº 14

Respuesta

El propietario de una aplicación creó un manifiesto de Pod mediante una `PersistentVolumeClaim` con un valor estándar de `StorageClassName`. ¿Qué tipo de almacenamiento se usa para este volumen en un clúster de GKE?

- A. Google Persistent Disk
- B. Volumen local en el nodo
- C. Con copia de seguridad de una memoria
- D. Almacenamiento de NFS



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](#)

Resumen

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.

