



Introduction to Kubernetes Workloads



Agenda

The kubectl command

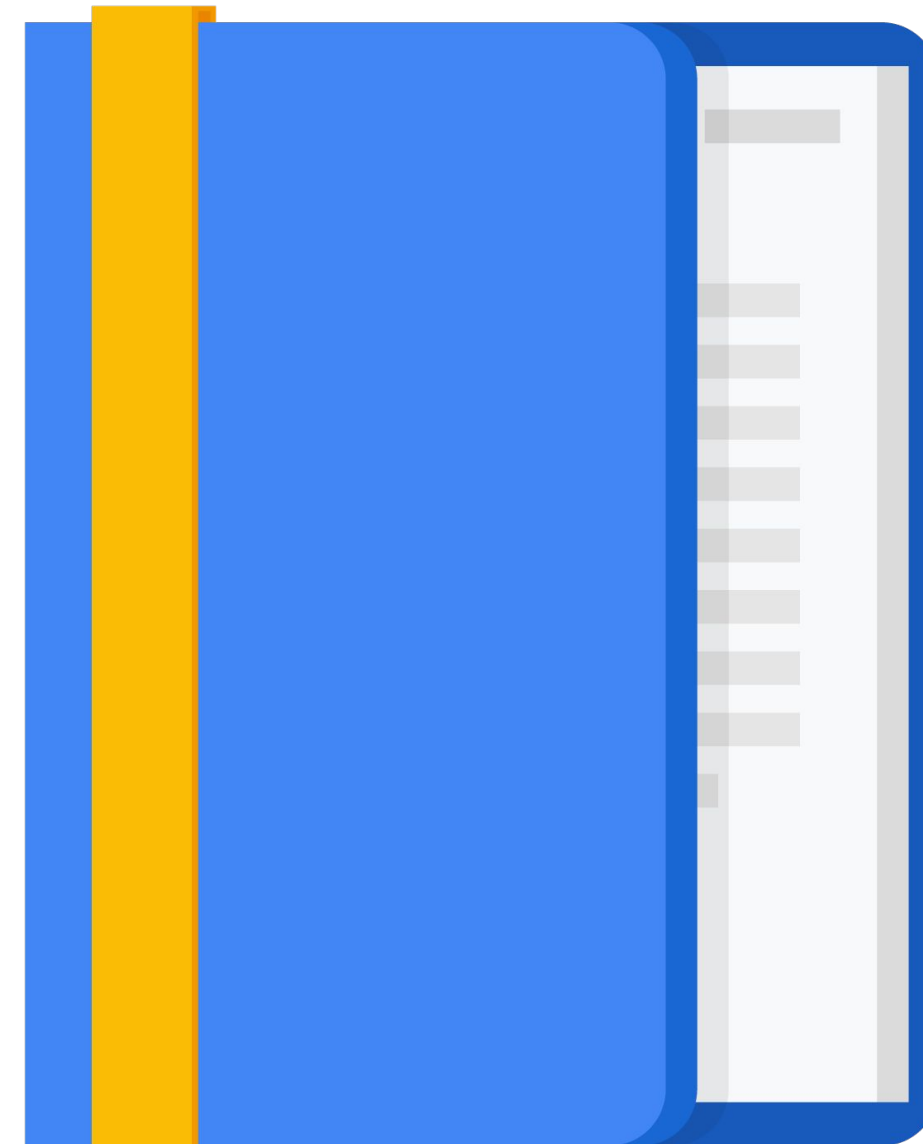
Deployments

Lab

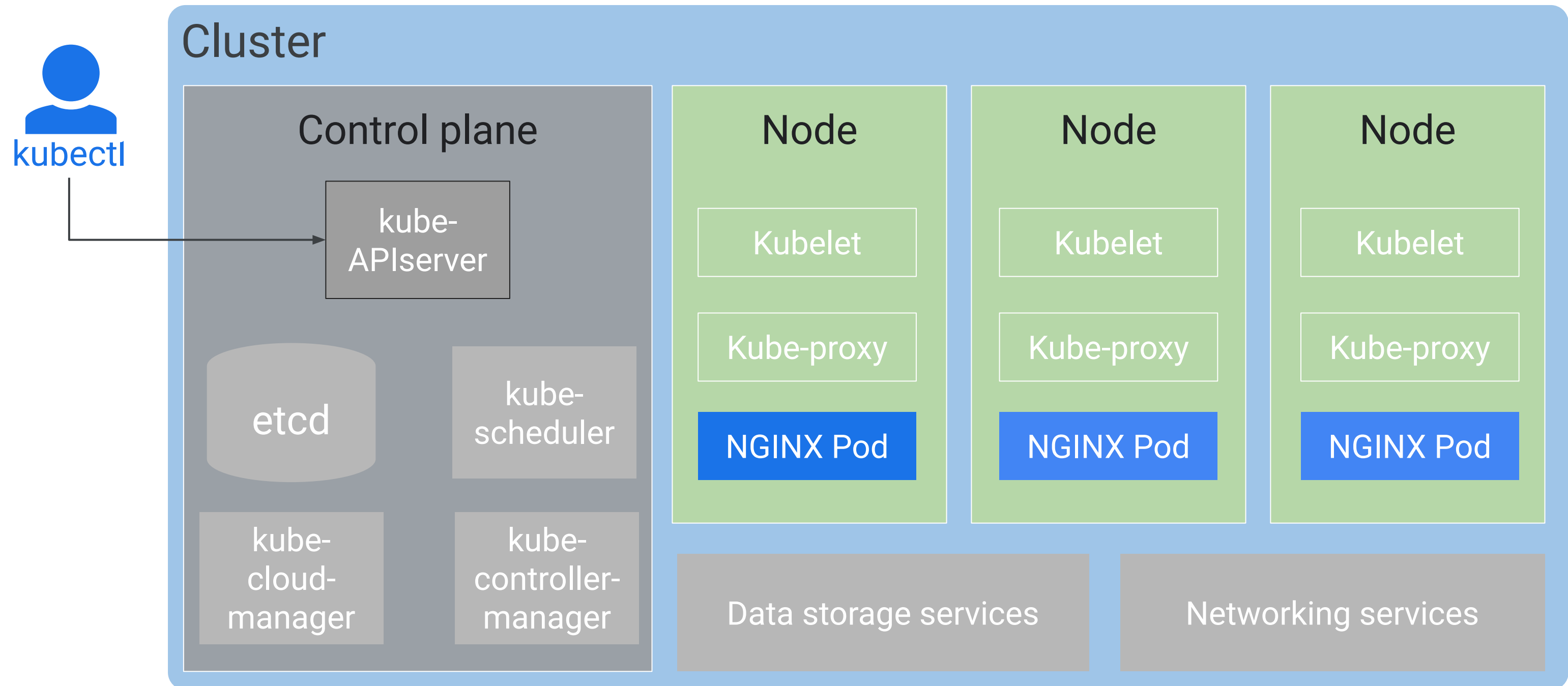
Kubernetes Controllers and Services

Quiz

Summary



Kubectl transforms your commands into API calls



The kubectl command has many uses

- Create Kubernetes objects
- View objects
- Delete objects
- View and export configurations

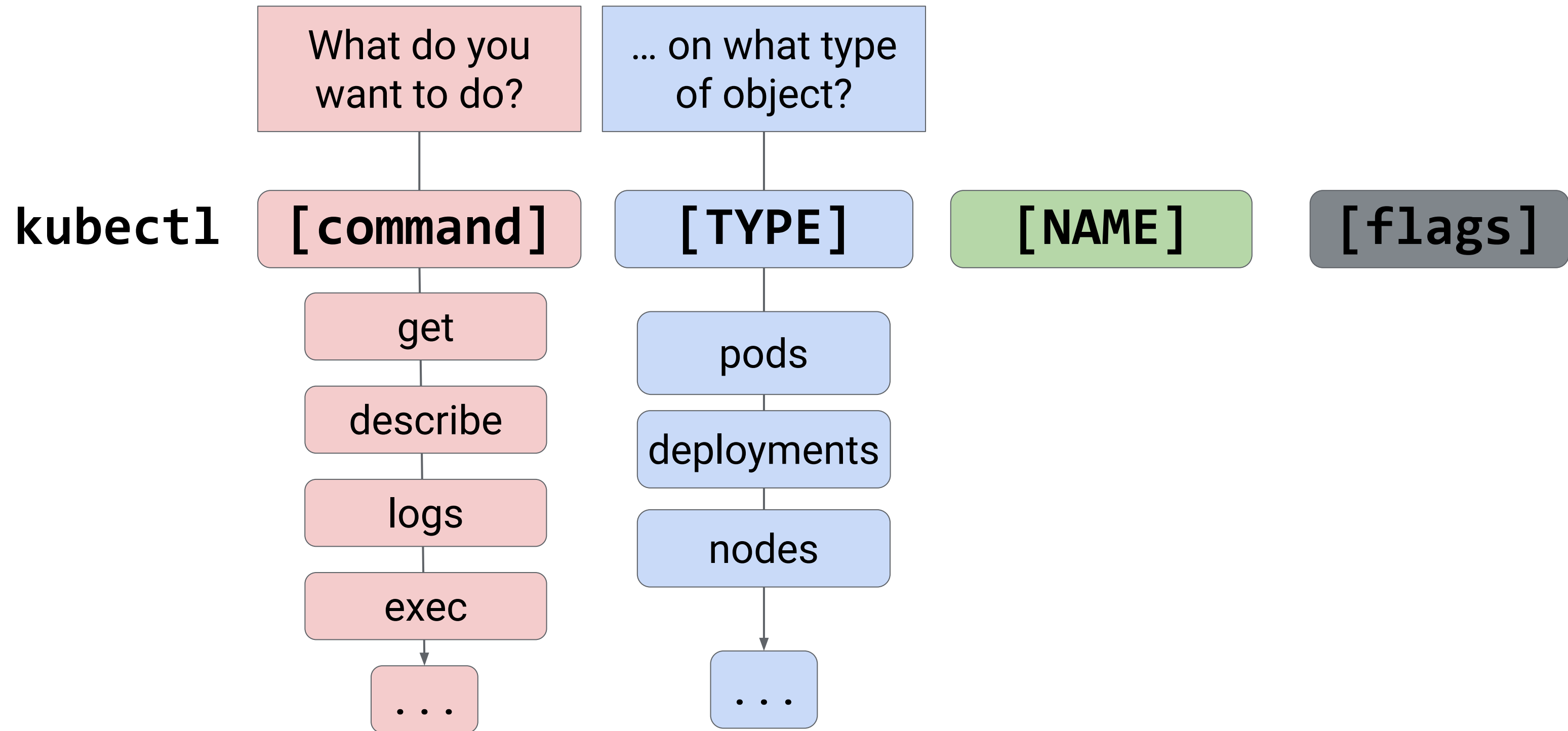
Example: connecting to a GKE cluster

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

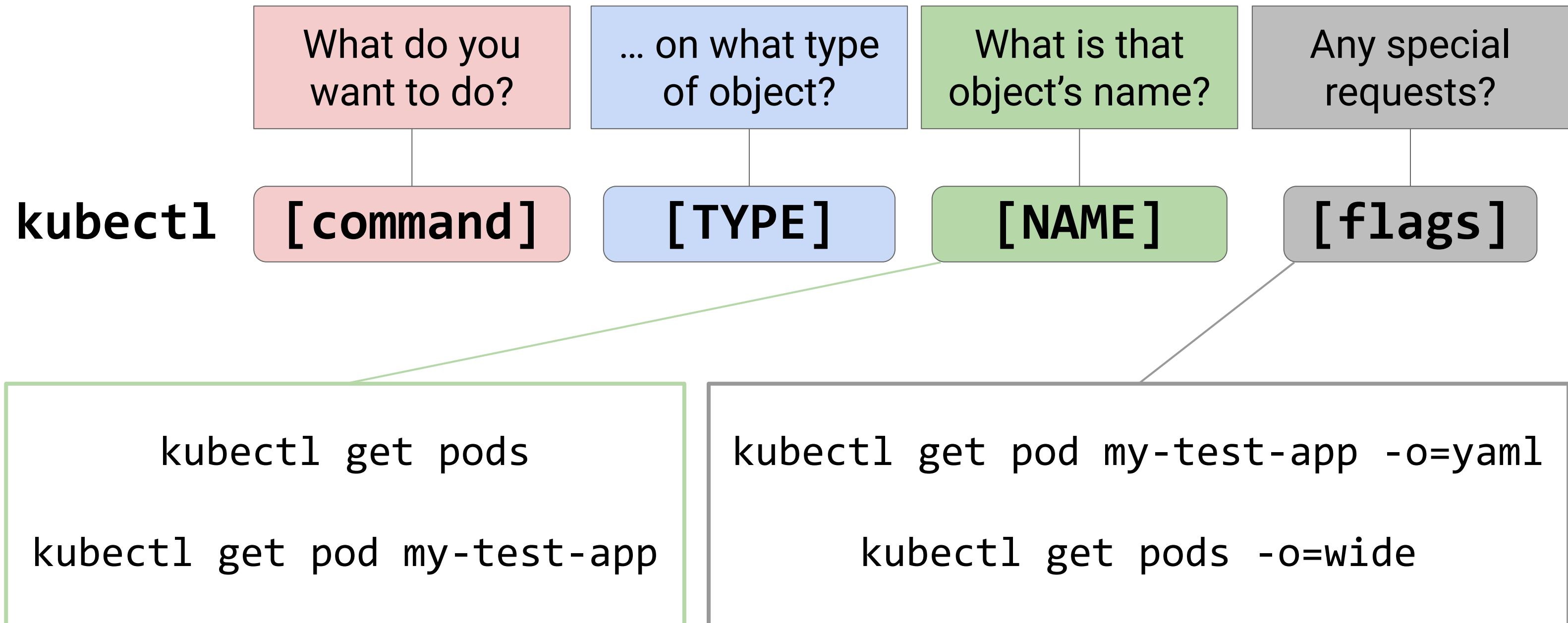
kubectl must be configured first

- Relies on a config file: `$HOME/.kube/config`.
- Config file contains:
 - Target cluster name
 - Credentials for the cluster
- Current config: `kubectl config view`.
- Sign in to a Pod interactively.

The kubectl command syntax has several parts



The kubectl command syntax has several parts



Agenda

The kubectl command

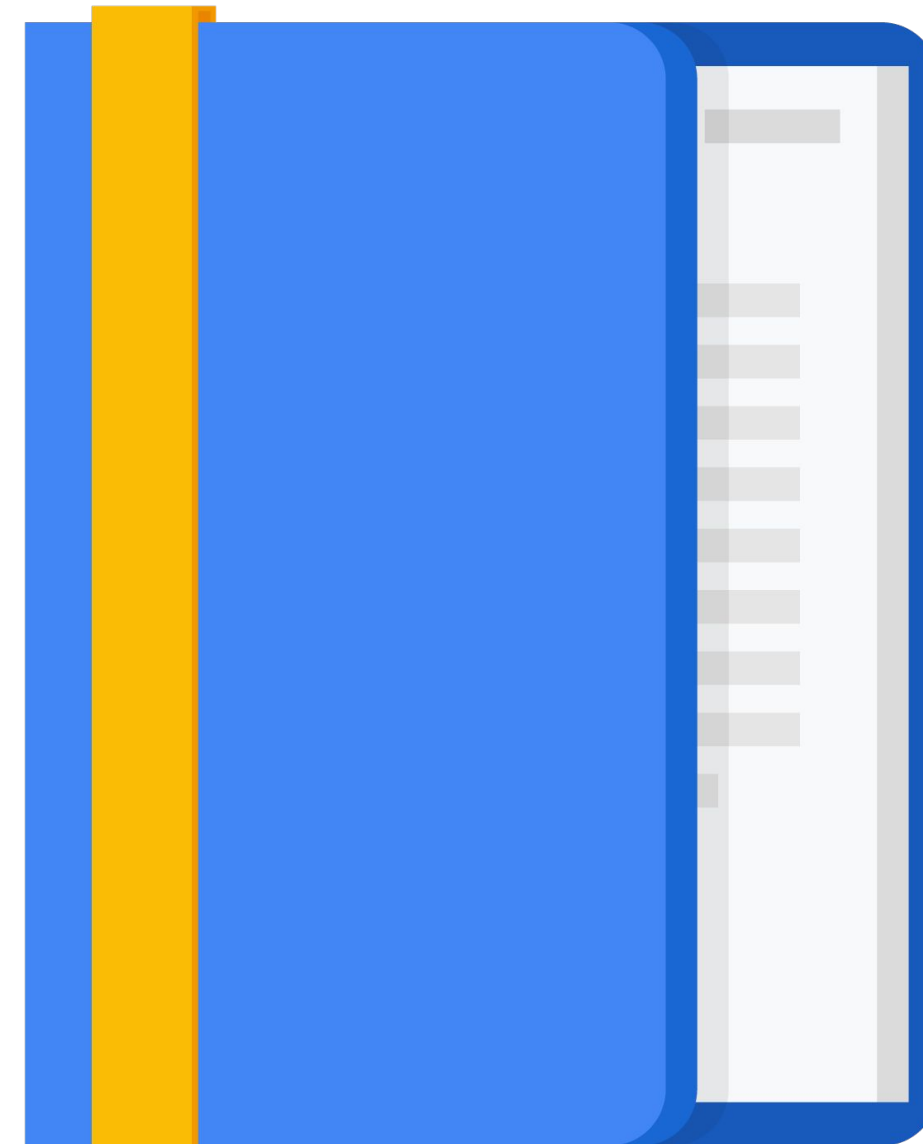
Deployments

Lab

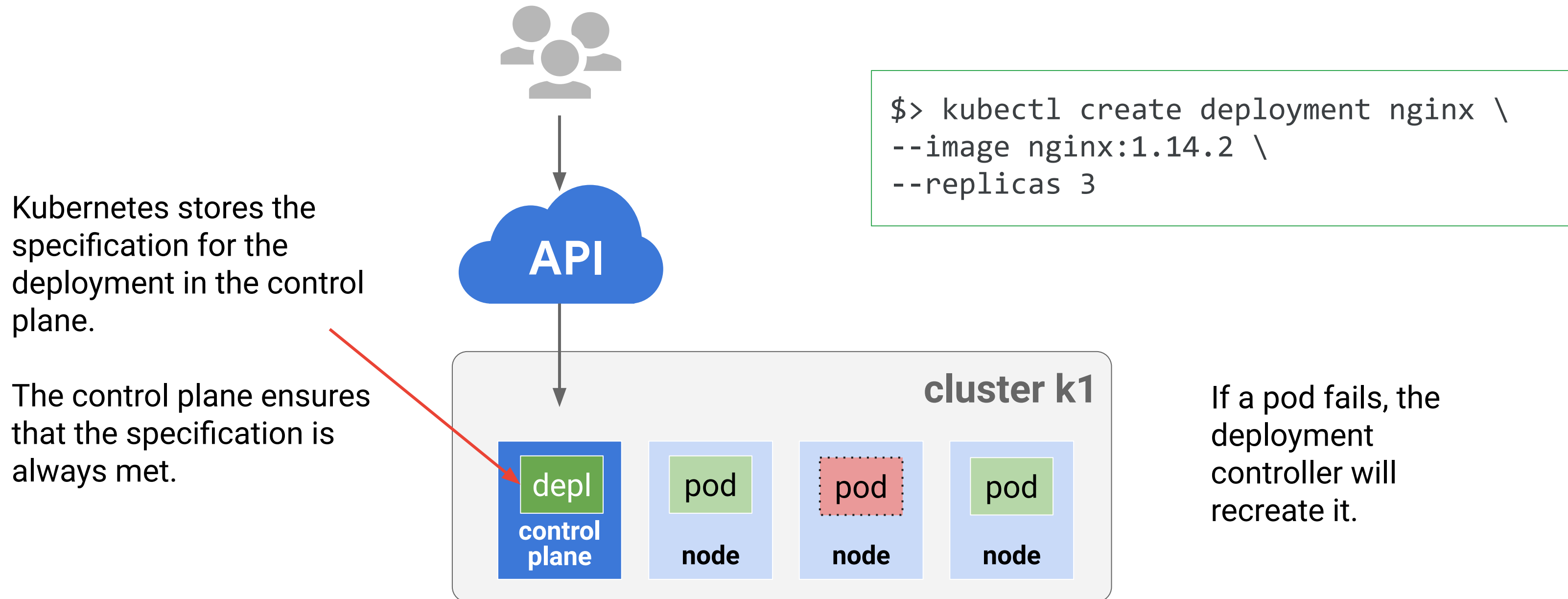
Kubernetes Controllers and Services

Quiz

Summary



A Deployment manages a set of replica Pods



There are three ways to create a 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
```

Deployment object file in YAML format

1

```
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
```

This is the preferred method for managing deployments in production.

The deployment YAML file should be placed under change control.

There are three ways to create a Deployment

← 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 *

app

Value

nginx-1

+ ADD KUBERNETES LABEL

3 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

Use `kubectl` to inspect your Deployment, or output the Deployment config in a YAML format

```
$ 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 the 'describe' command to get detailed info

```
$ 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
```


Or use the Cloud Console

✓ nginx-deployment

1

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

OVERVIEW

DETAILS

REVISION HISTORY

EVENTS

LOGS

YAML

CPU ?

Memory ?

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

You can scale the Deployment manually

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

ACTIONS ▾ KUBE

- Autoscale
- Expose
- Rolling update
- Scale**
- Automated deployment

Edit default-pool

Node version
1.20.10-gke.301
[CHANGE](#)

Size
Number of nodes *
2

☐ Enable autoscaling ?

You can also autoscale the 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](#)

You can update a Deployment in different ways

```
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]
```

You can update a Deployment in different ways

[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

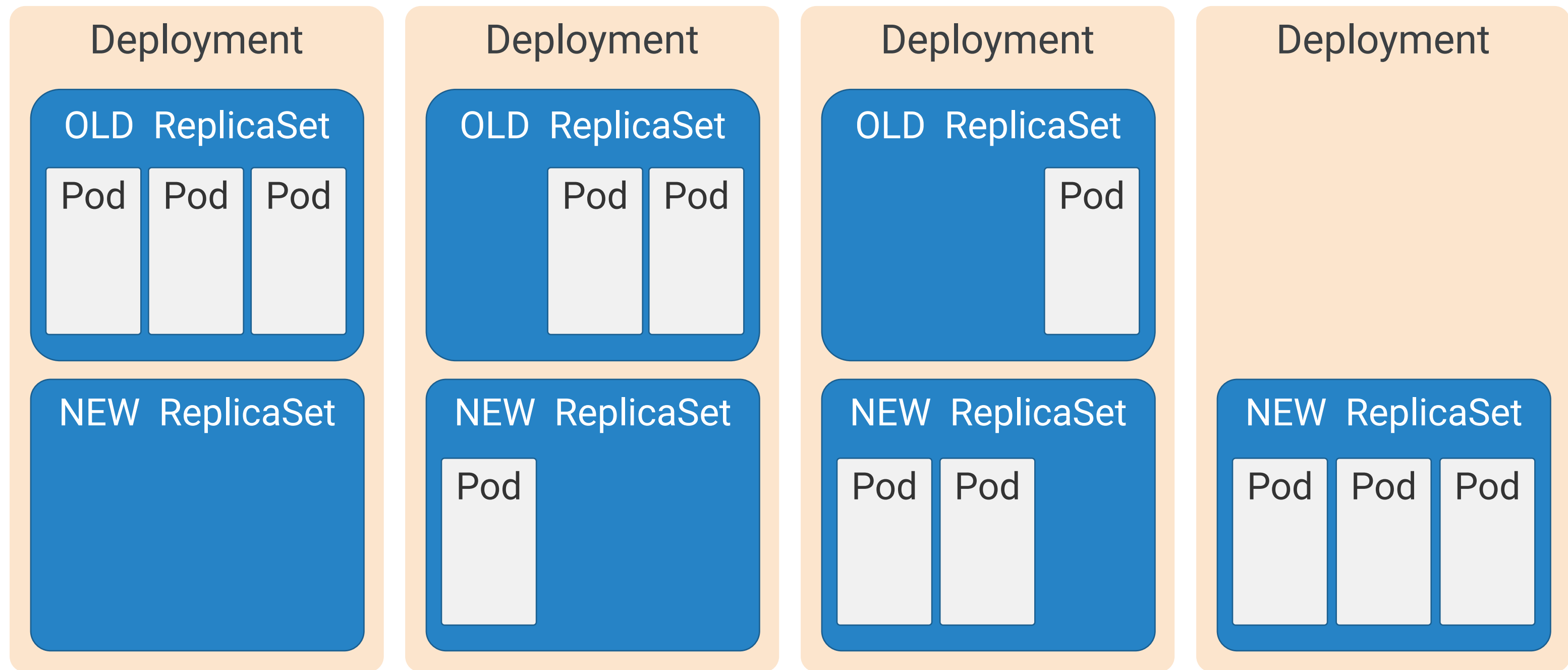
Image of nginx *

nginx:1.7.9

* Indicates required field

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

The process behind updating a Deployment



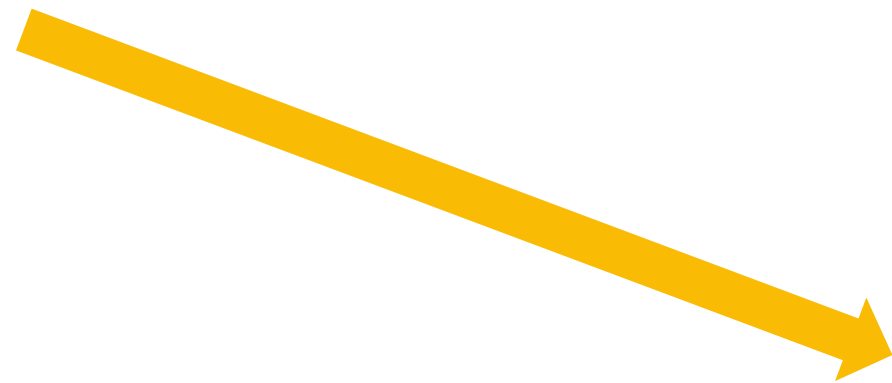
Evaluation Time!



Course Evaluation: we need your feedback!

1 Go to the labs page for the class

2 Click the Google Cloud Training Survey link

A screenshot of the Google Cloud Fundamentals: Core Infrastructure Labs page. The page has a blue header with a back arrow and the text "Google Cloud Fundamentals: Core Infrastructure". Below the header, there are two tabs: "Labs" (which is selected and underlined) and "Lecture Notes". The "Labs" tab displays a list of seven labs, each with a warning icon (a triangle with an exclamation mark) and the text "Google Cloud Fundamentals: Getting Started with [Service Name]". The services listed are Cloud Marketplace, Compute Engine, Cloud Storage and Cloud SQL, GKE, App Engine, Deployment Manager and Cloud Monitoring, and BigQuery. At the bottom of the page, there is a blue link that says "Take class evaluation: Google Cloud Training Survey".

← Google Cloud Fundamentals: Core Infrastructure

Labs Lecture Notes

- ⚠ Google Cloud Fundamentals: Getting Started with Cloud Marketplace
- ⚠ Google Cloud Fundamentals: Getting Started with Compute Engine
- ⚠ Google Cloud Fundamentals: Getting Started with Cloud Storage and Cloud SQL
- ⚠ Google Cloud Fundamentals: Getting Started with GKE
- ⚠ Google Cloud Fundamentals: Getting Started with App Engine
- ⚠ Google Cloud Fundamentals: Getting Started with Deployment Manager and Cloud Monitoring
- ⚠ Google Cloud Fundamentals: Getting Started with BigQuery

Take class evaluation: [Google Cloud Training Survey](#)

Lab Intro

Creating Google Kubernetes
Engine Deployments

Duration: 30 minutes



Agenda

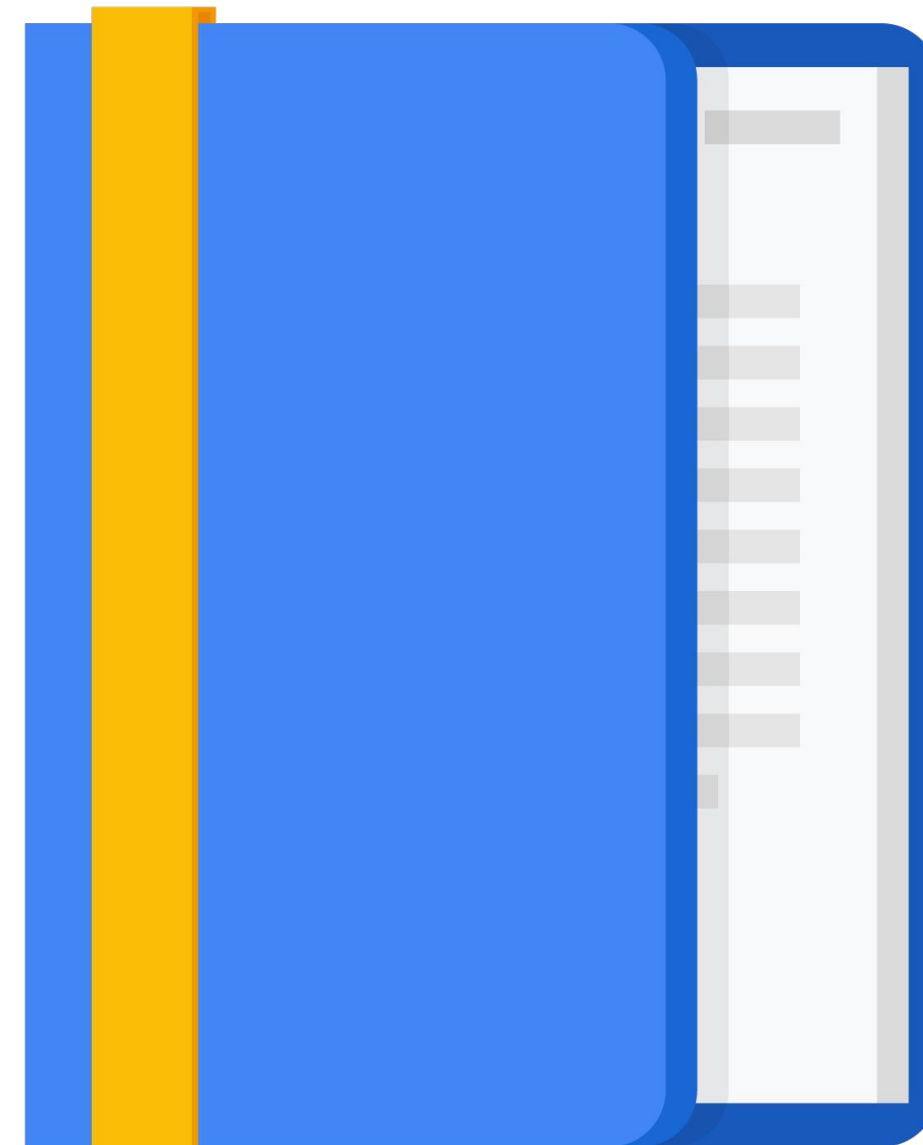
The kubectl command

Deployments

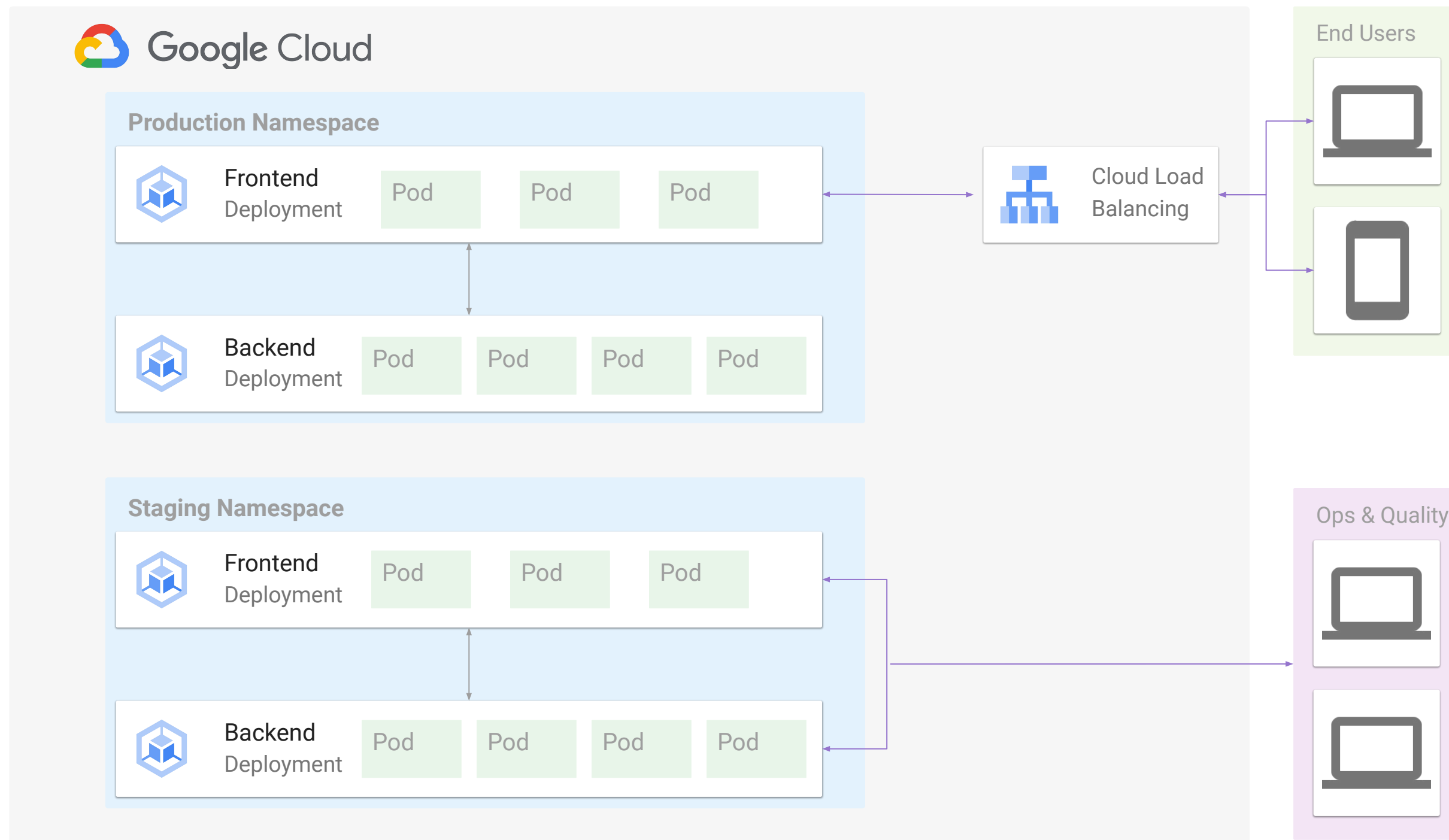
Lab

Kubernetes Controllers and Services

Quiz



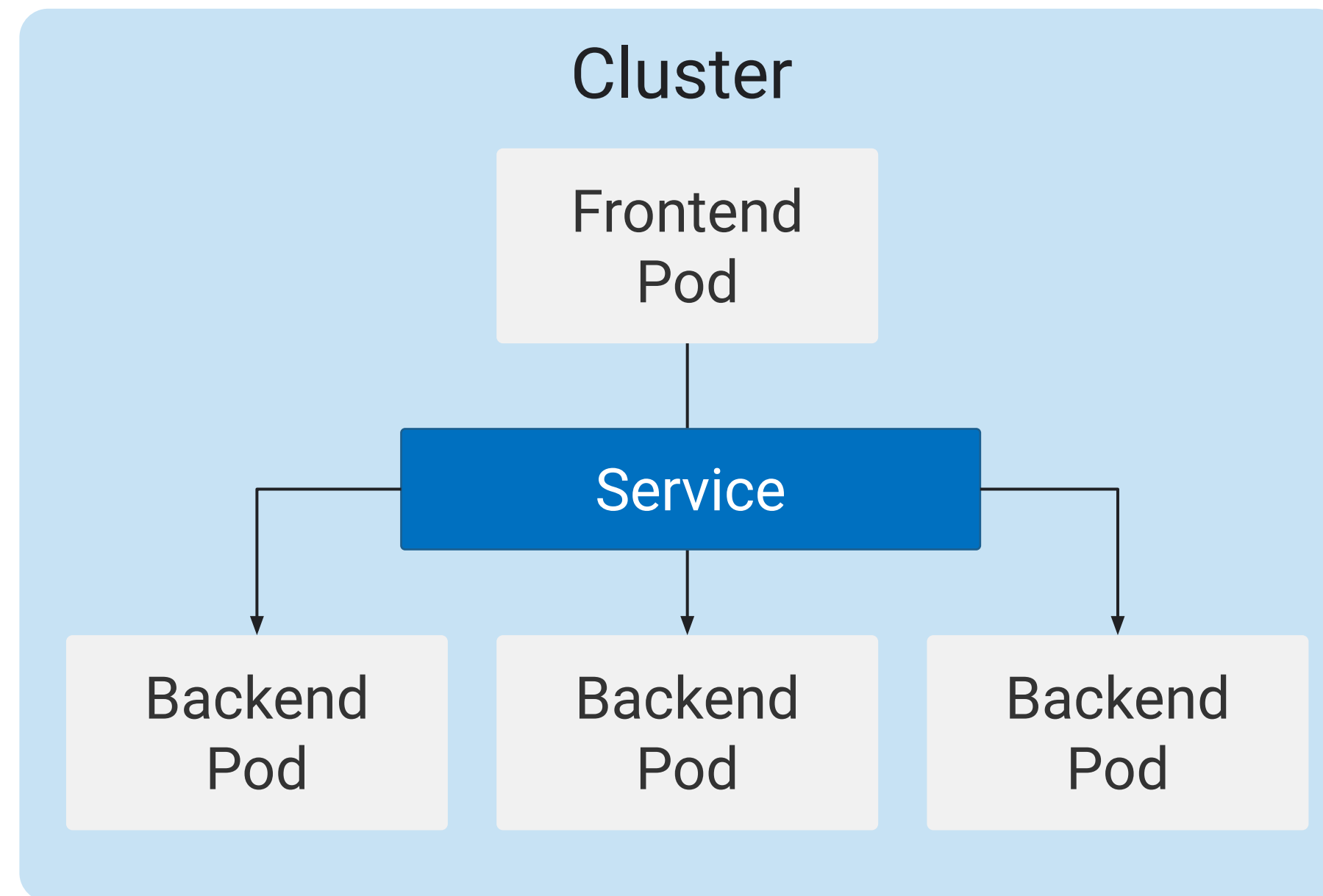
Your workload doesn't run in a single Pod



A Deployment is one way to run an application

Kubernetes Controller	Description	Use Case
Deployment	manages a set of stateless pods that run continuously	Typical for stateless web servers and application servers
StatefulSet	manages a set of pods with attached storage that run continuously	For database servers and other stateful servers
Job	manages a set of pods that run to task completion and then exit	Suited to applications that execute a work queue
CronJob	Like a Job controller, but runs pods on a schedule	

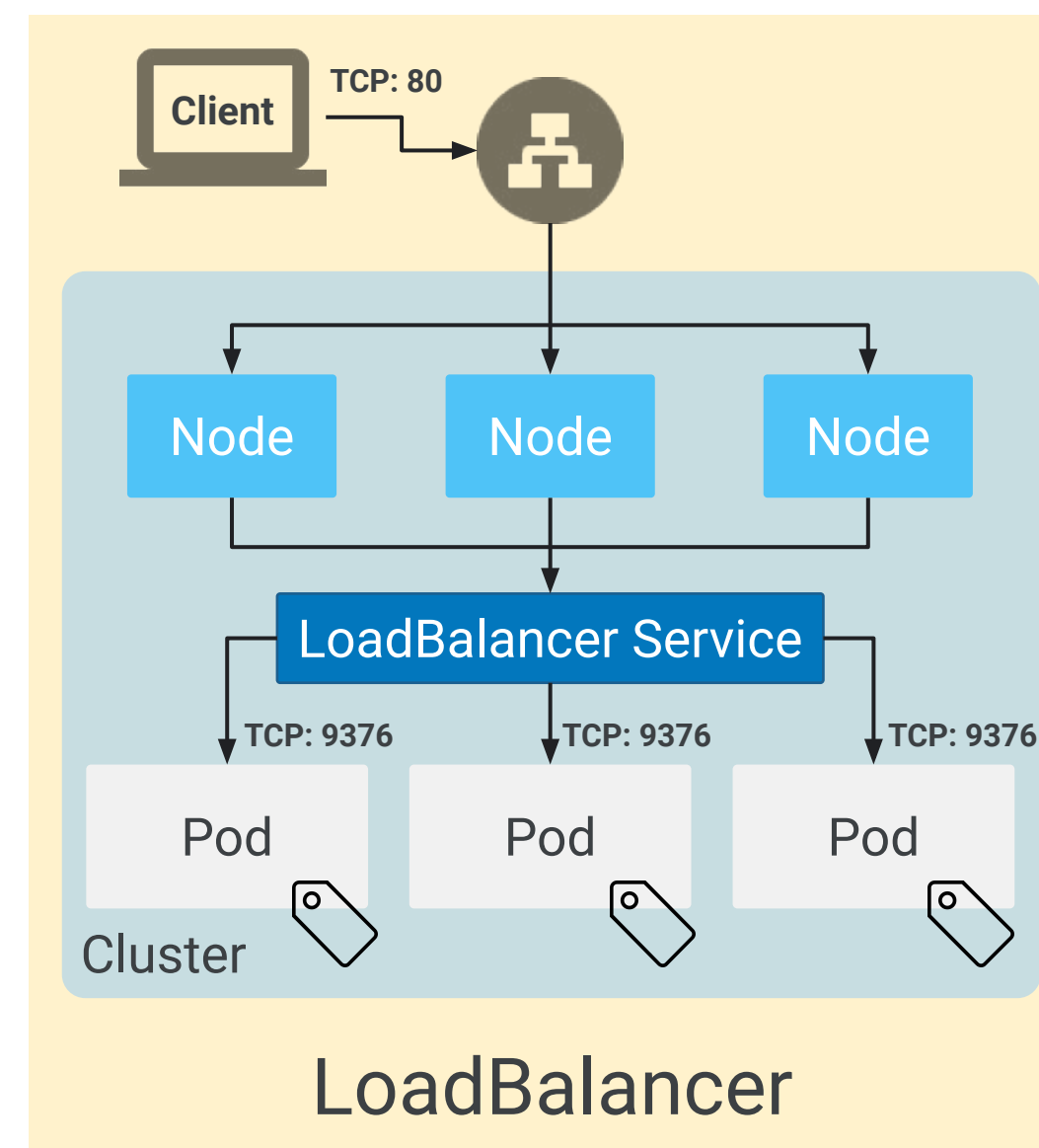
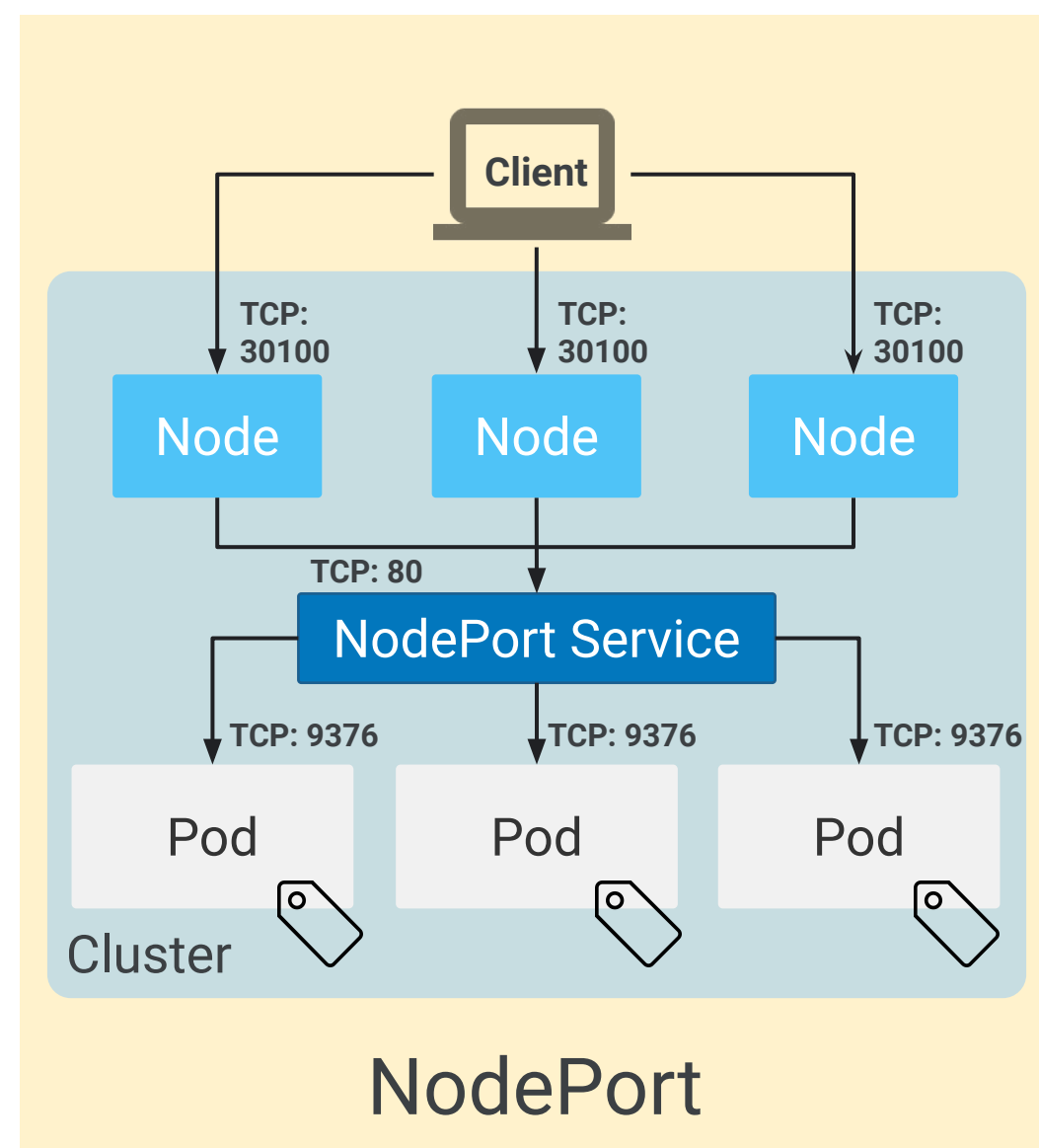
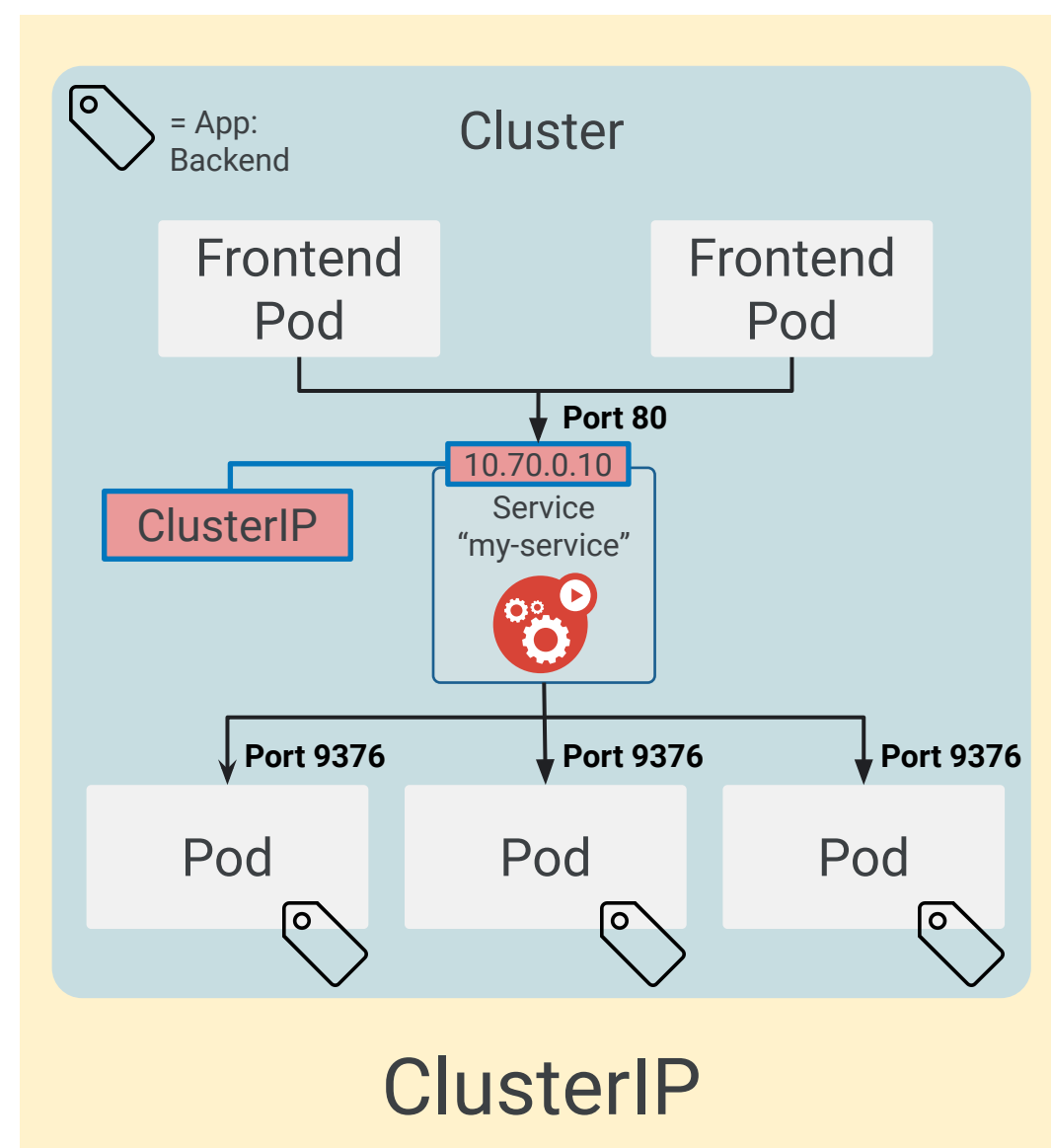
Service is a stable network representation of a set of pods



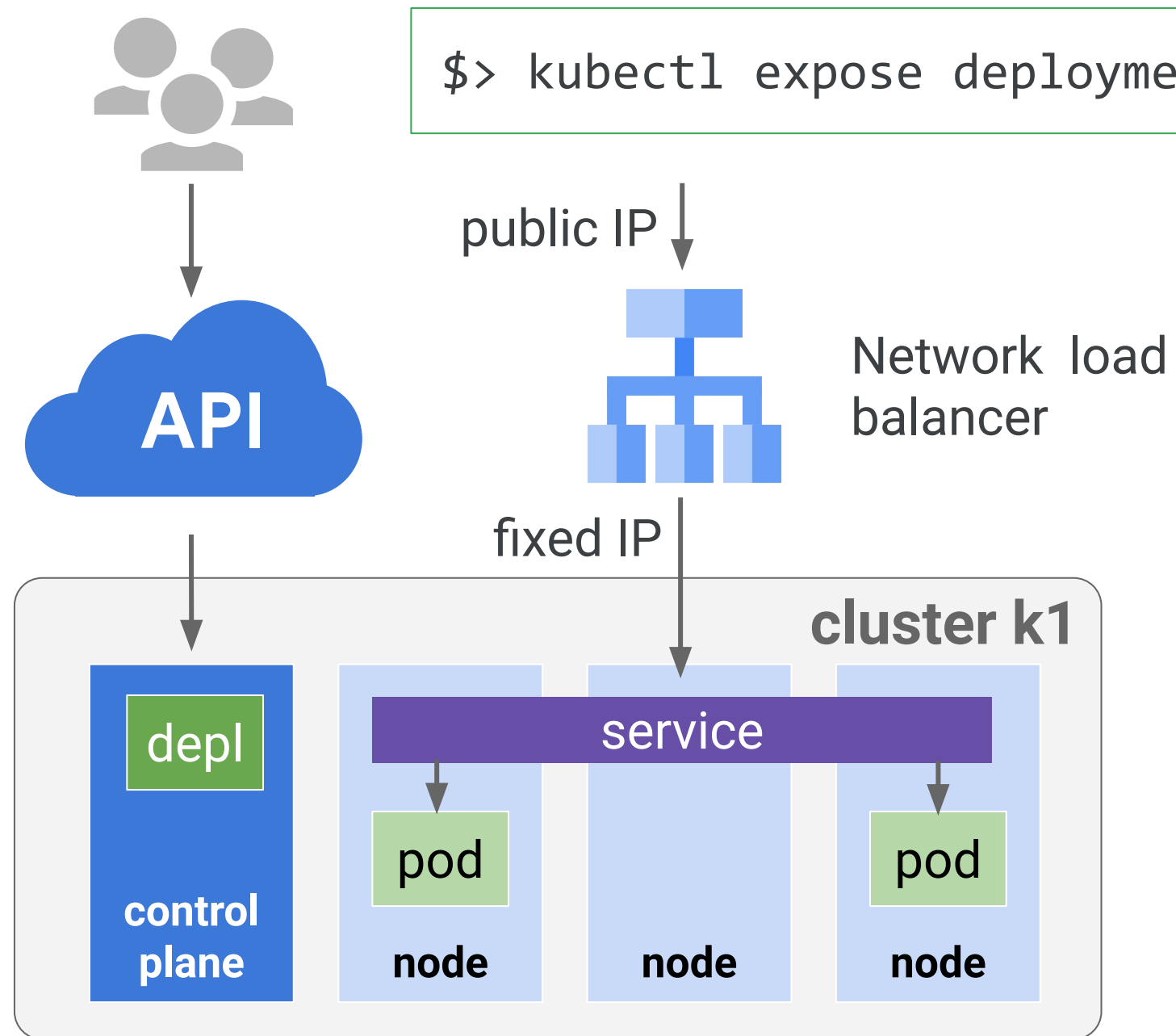
Kubernetes Service Types

Type	Description	
LoadBalancer	Exposes the group of pods externally using a load balancer	Can be accessed from outside the cluster using the load balancer's IP Very common service
NodePort	Exposes the group of pods on each node's IP at a static port	Can be accessed from outside the cluster using <NodeIP>:<NodePort>
ClusterIP	Exposes an internal IP address for a group of pods	Cannot be accessed from outside the cluster

Service types summary



Creating a load balancer service from the CLI



Creating a LoadBalancer Service

```
apiVersion: v1
kind: Service
metadata:
  name: my-app-lb
spec:
  type: LoadBalancer
  selector:
    app: my-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

Agenda

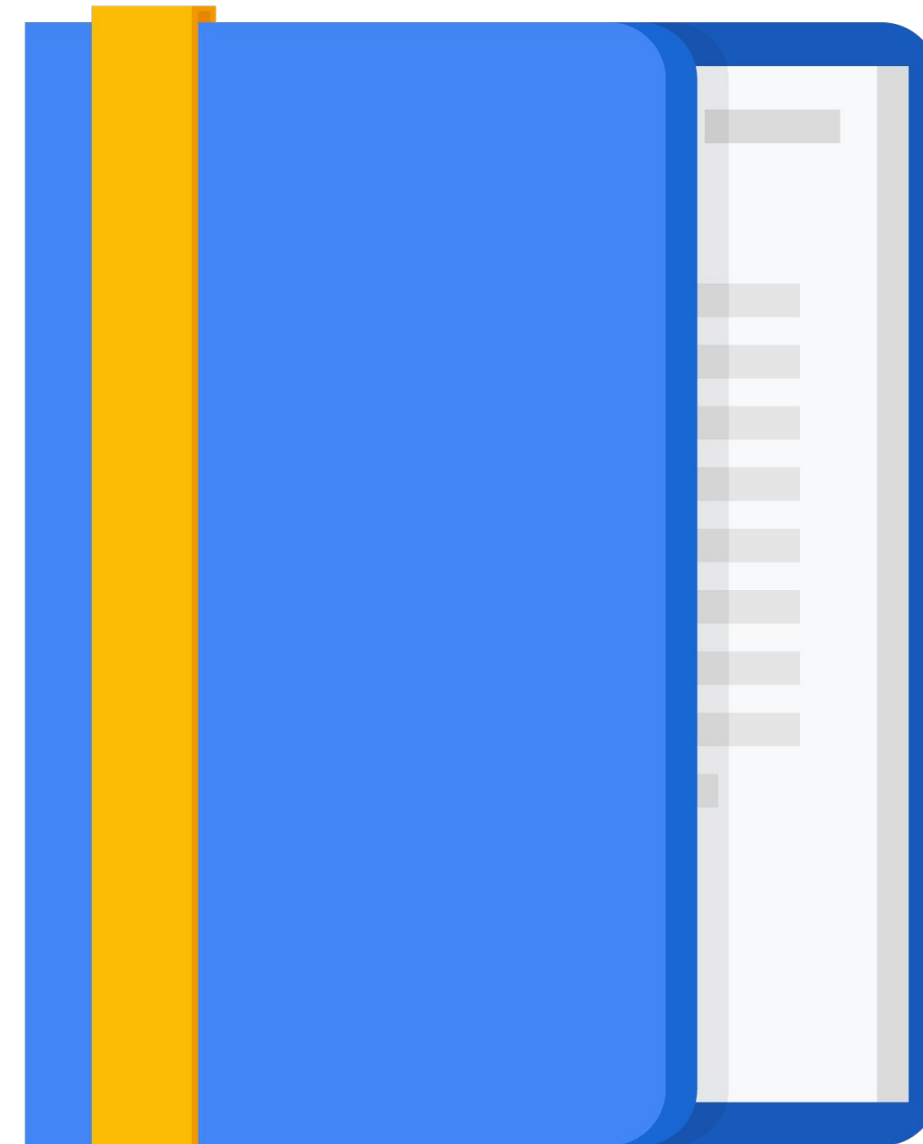
The kubectl command

Deployments

Lab

Kubernetes Controllers and Services

Quiz



Question #1

Question

You want to use a `kubectl get` command to identify which Node each Pod is running on. Which command do you need to execute?

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

Question #1

Answer

You want to use a `kubectl get` command to identify which Node each Pod is running on. Which command do you need to execute?

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



Question #2

Question

After a Deployment has been created and its component Pods are running, which component is responsible for ensuring that a replacement Pod is launched whenever a Pod fails or is evicted?

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

Question #2

Answer

After a Deployment has been created and its component Pods are running, which component is responsible for ensuring that a replacement Pod is launched whenever a Pod fails or is evicted?

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



Question #3

Question

What is the relationship between Deployments and ReplicaSets?

- A. A Deployment configures a ReplicaSet controller to create and maintain a specific version of the Pods that the Deployment specifies.
- B. A Deployment configures a ReplicaSet controller to create and maintain all the Pods that the Deployment specifies, regardless of their version.
- C. A ReplicaSet configures a Deployment controller to create and maintain a specific version of the Pods that the Deployment specifies.
- D. There is no relationship; in modern Kubernetes, Replication Controllers are typically used to maintain a set of Pods in a running state.

Question #3

Answer

What is the relationship between Deployments and ReplicaSets?

- A. A Deployment configures a ReplicaSet controller to create and maintain a specific version of the Pods that the Deployment specifies.
- B. A Deployment configures a ReplicaSet controller to create and maintain all the Pods that the Deployment specifies, regardless of their version.
- C. A ReplicaSet configures a Deployment controller to create and maintain a specific version of the Pods that the Deployment specifies.
- D. There is no relationship; in modern Kubernetes, Replication Controllers are typically used to maintain a set of Pods in a running state.



Question #4

Question

What type of application is suited for use with a Deployment?

- A. Batch
- B. Stateful
- C. Stateless
- D. Written in Go

Question #4

Answer

What type of application is suited for use with a Deployment?

- A. Batch
- B. Stateful
- C. Stateless
- D. Written in Go



Question #5

Question

In GKE, what is the source of the IP addresses for Pods?

- A. Address ranges assigned to your Virtual Private Cloud
- B. Arbitrary network addresses per cluster
- C. Loopback network addresses

Question #5

Answer

In GKE, what is the source of the IP addresses for Pods?

- A. Address ranges assigned to your Virtual Private Cloud
- B. Arbitrary network addresses per cluster
- C. Loopback network addresses



Question #6

Question

Your Pod has been rescheduled and the IP address that was assigned to the Pod when it was originally scheduled is no longer accessible. What is the reason for this?

- A. The new Pod IP address is blocked by a firewall.
- B. The new Pod has received a different IP address.
- C. The old Pod IP address is blocked by a firewall.
- D. The Pod IP range for the cluster is exhausted.

Question #6

Answer

Your Pod has been rescheduled and the IP address that was assigned to the Pod when it was originally scheduled is no longer accessible. What is the reason for this?

- A. The new Pod IP address is blocked by a firewall.
- B. The new Pod has received a different IP address.
- C. The old Pod IP address is blocked by a firewall.
- D. The Pod IP range for the cluster is exhausted.



