

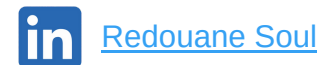


Kubernetes for Newbies



A fundamental overview of why adopting Kubernetes is beneficial, including an explanation of its operation and core architecture.

Presented by: Redouane Soul



Monolithic App



Home



My Network



Jobs



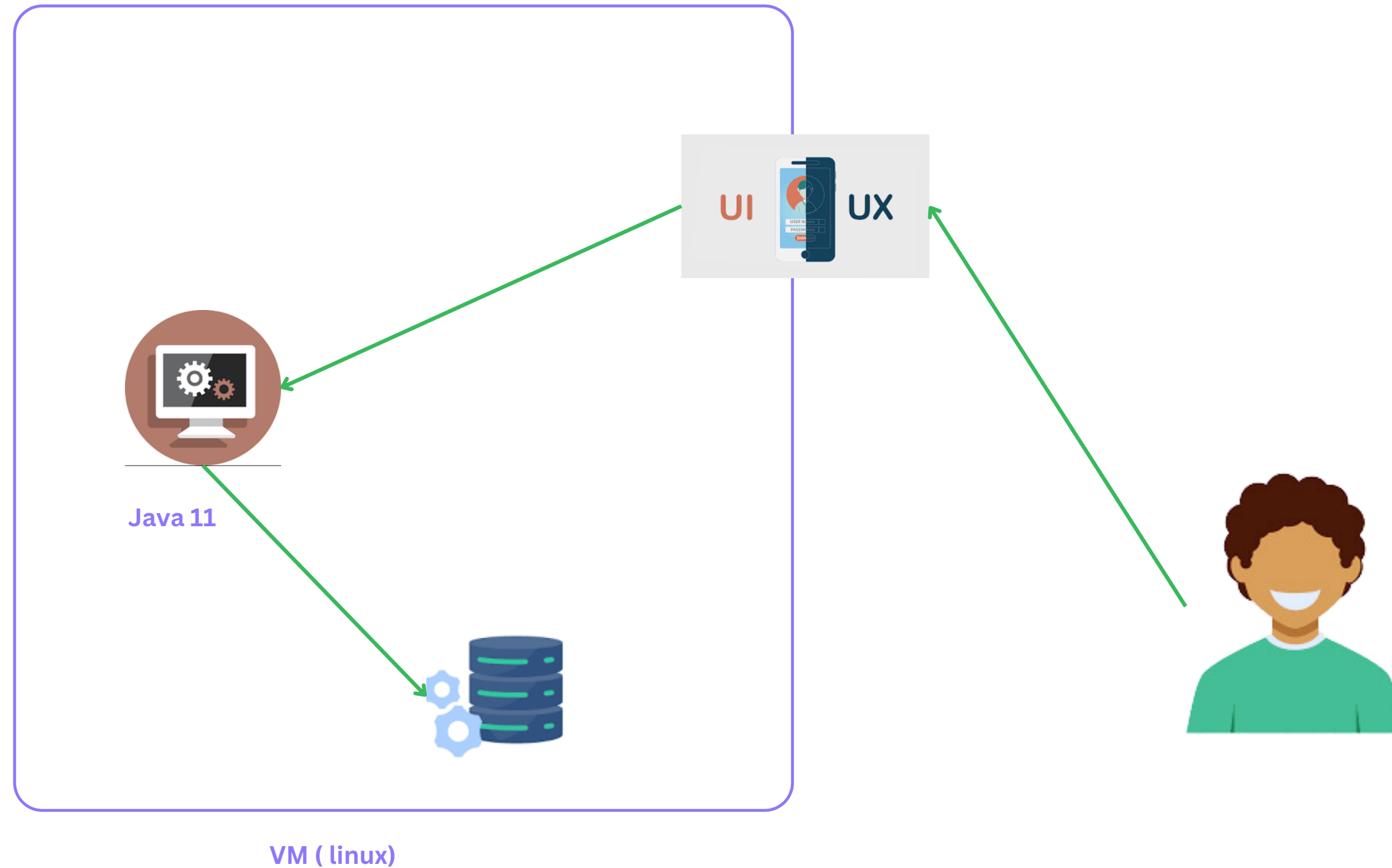
Messaging



Notifications

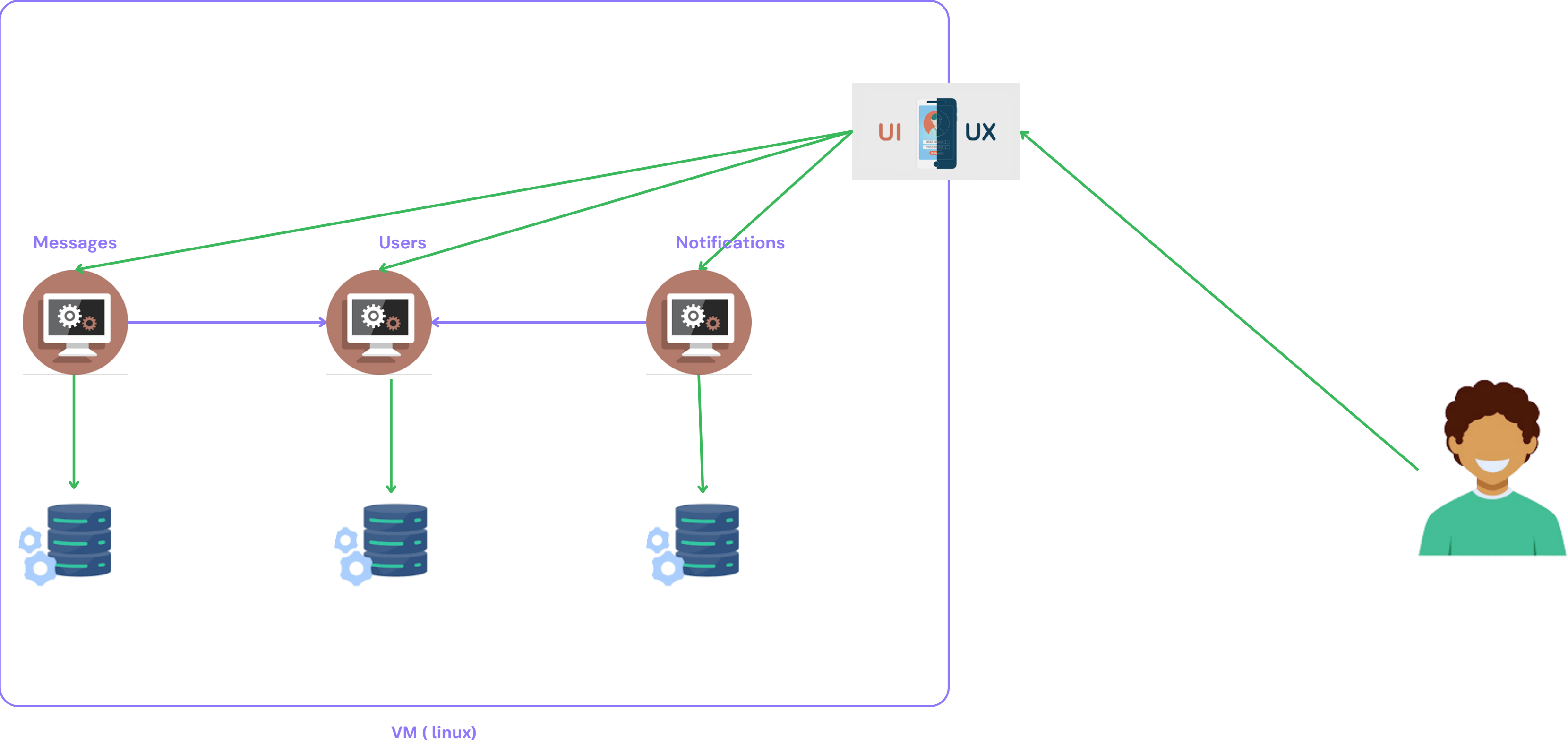


Me ▼

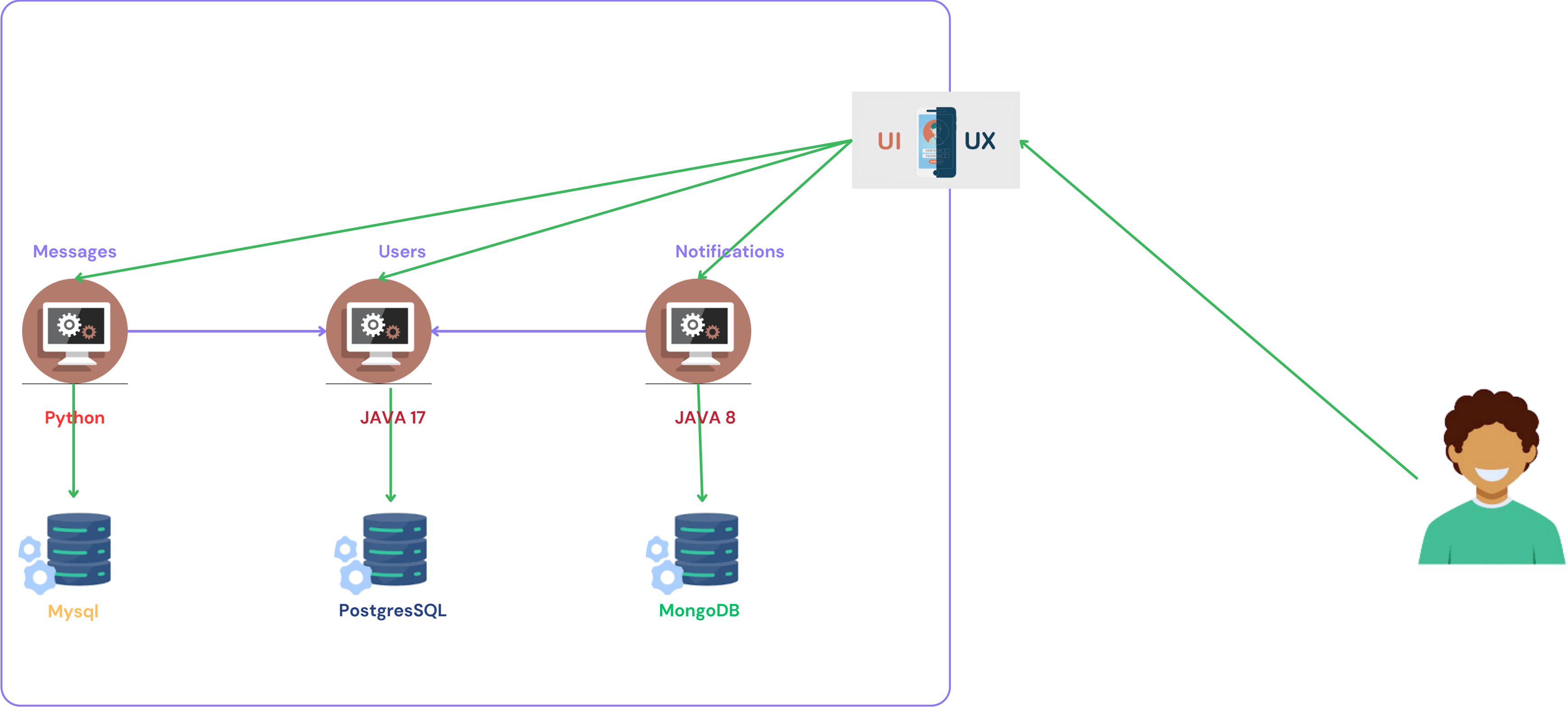


- Slower development speed
- Scalability
- Deployment

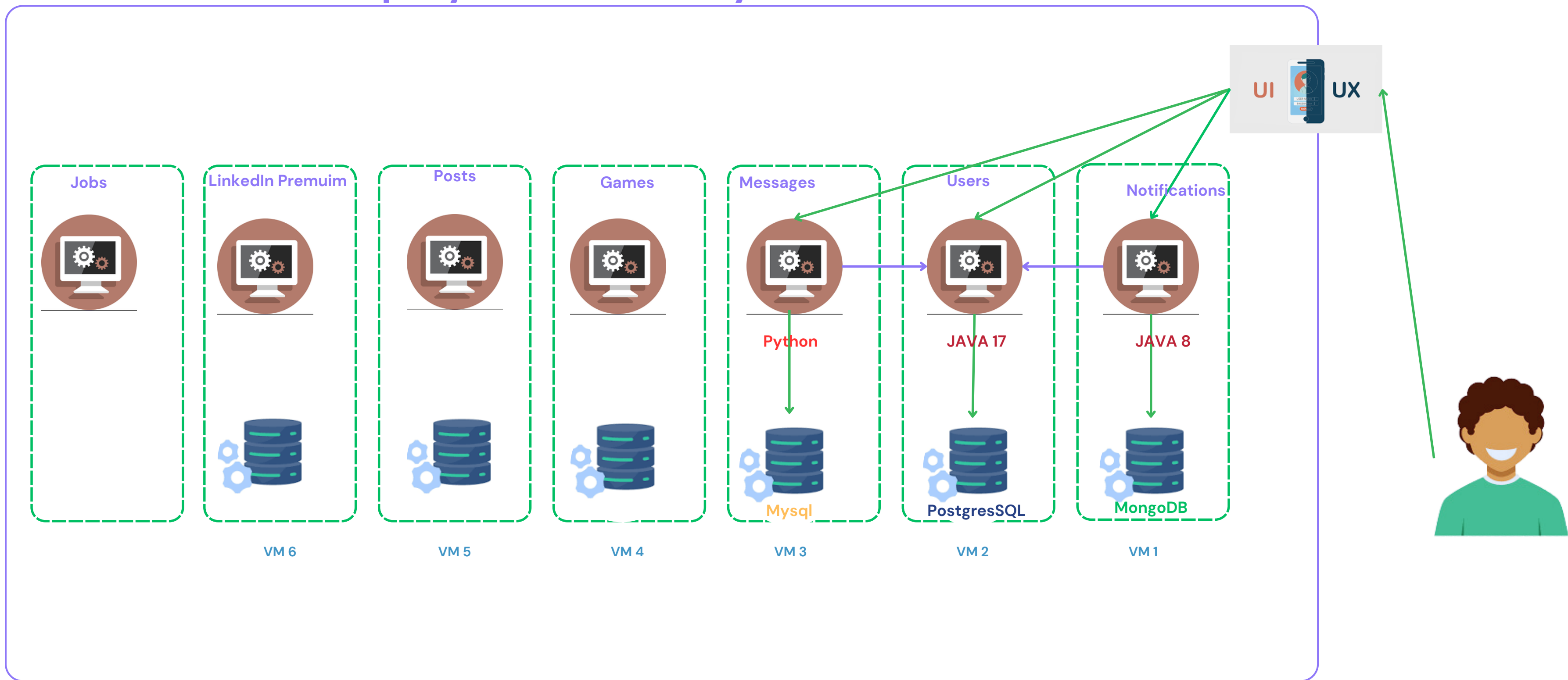
Microservices App



Microservices App



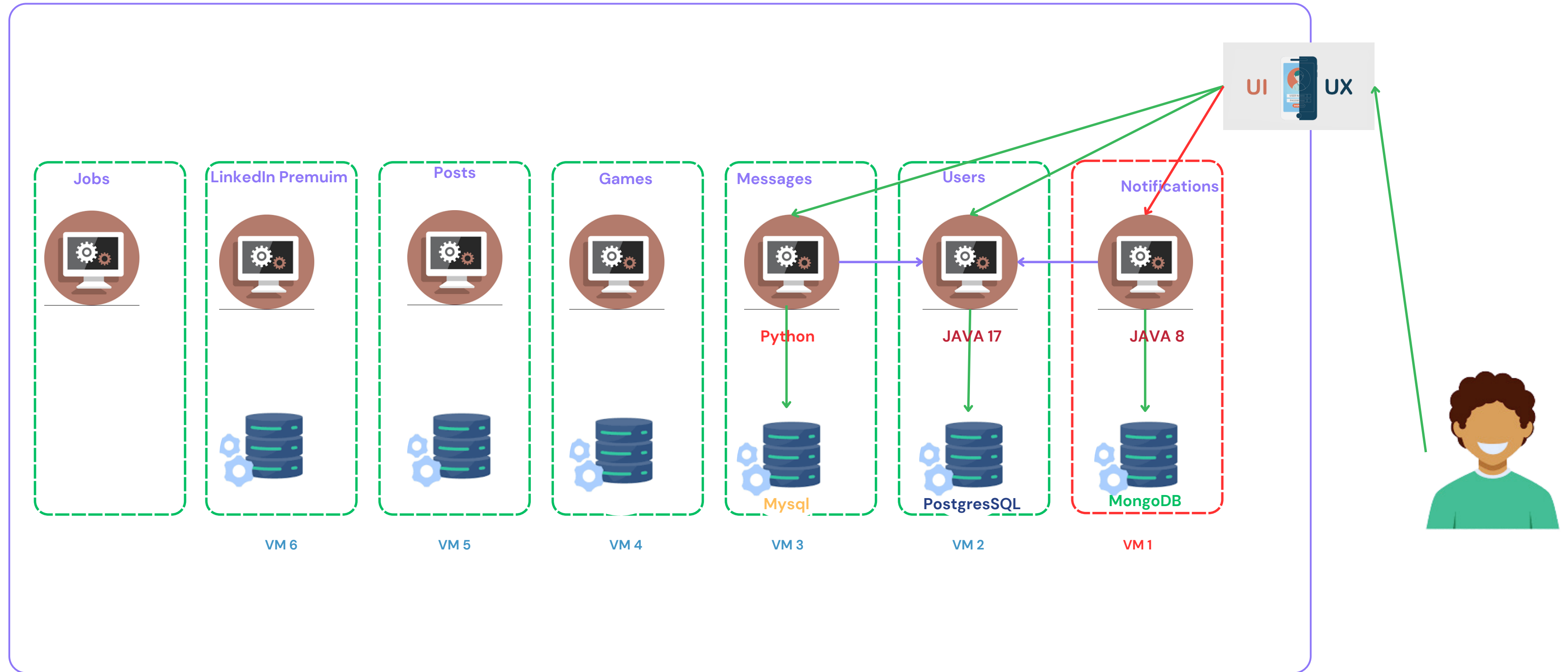
Hard to deploy and monitor microservices



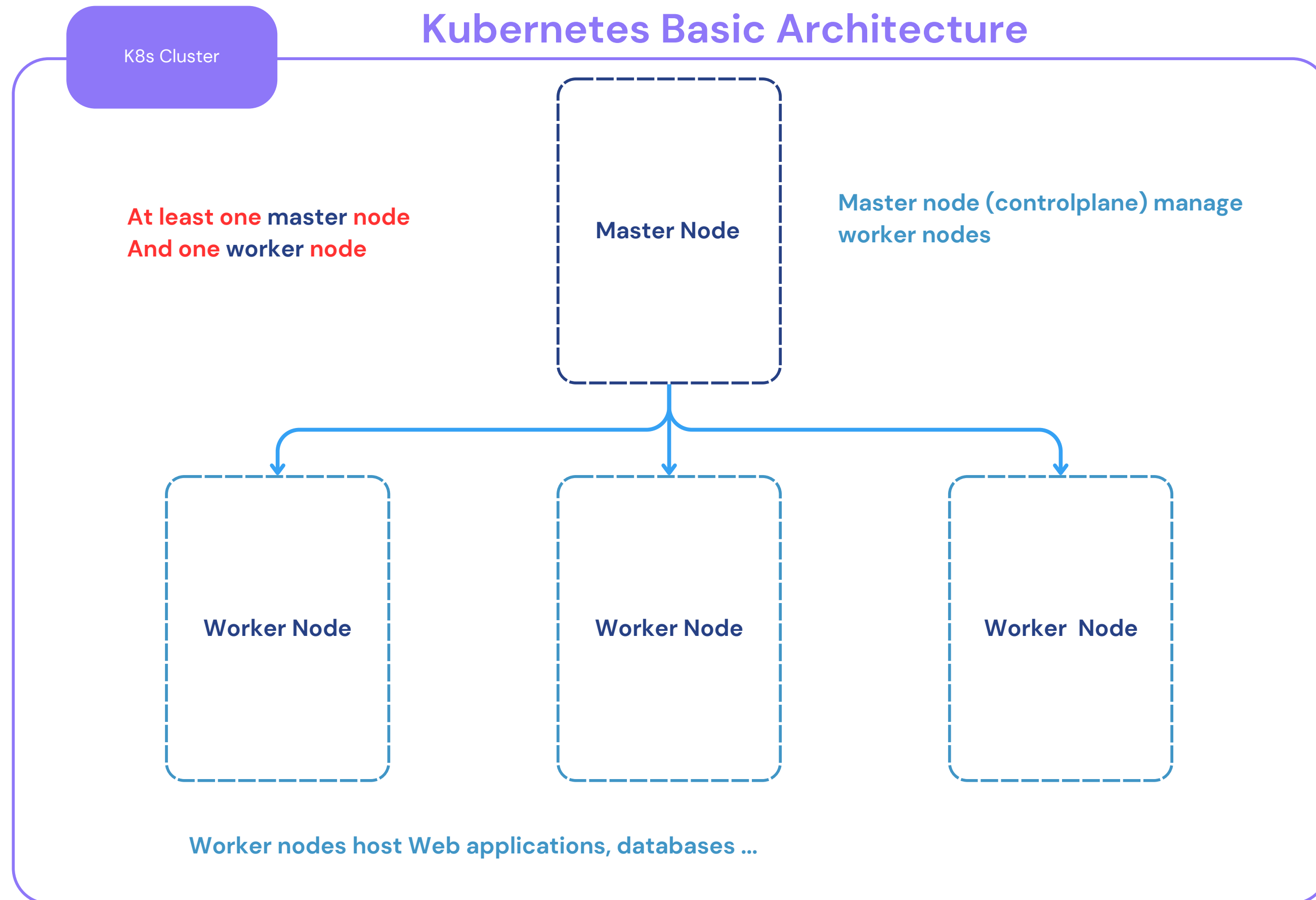
- Self Healing

- Scalability

- Disaster Recovery

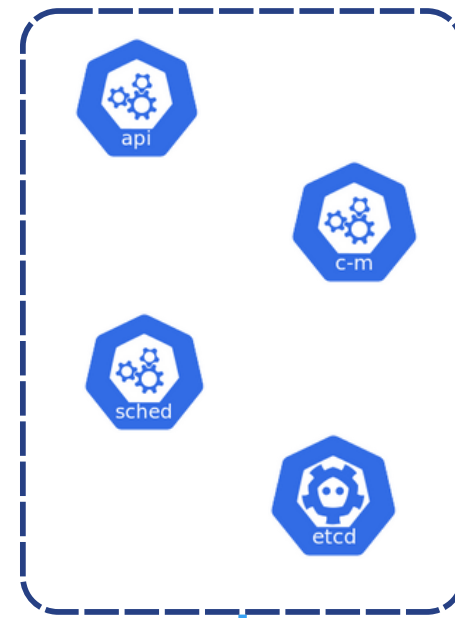


Kubernetes Basic Architecture

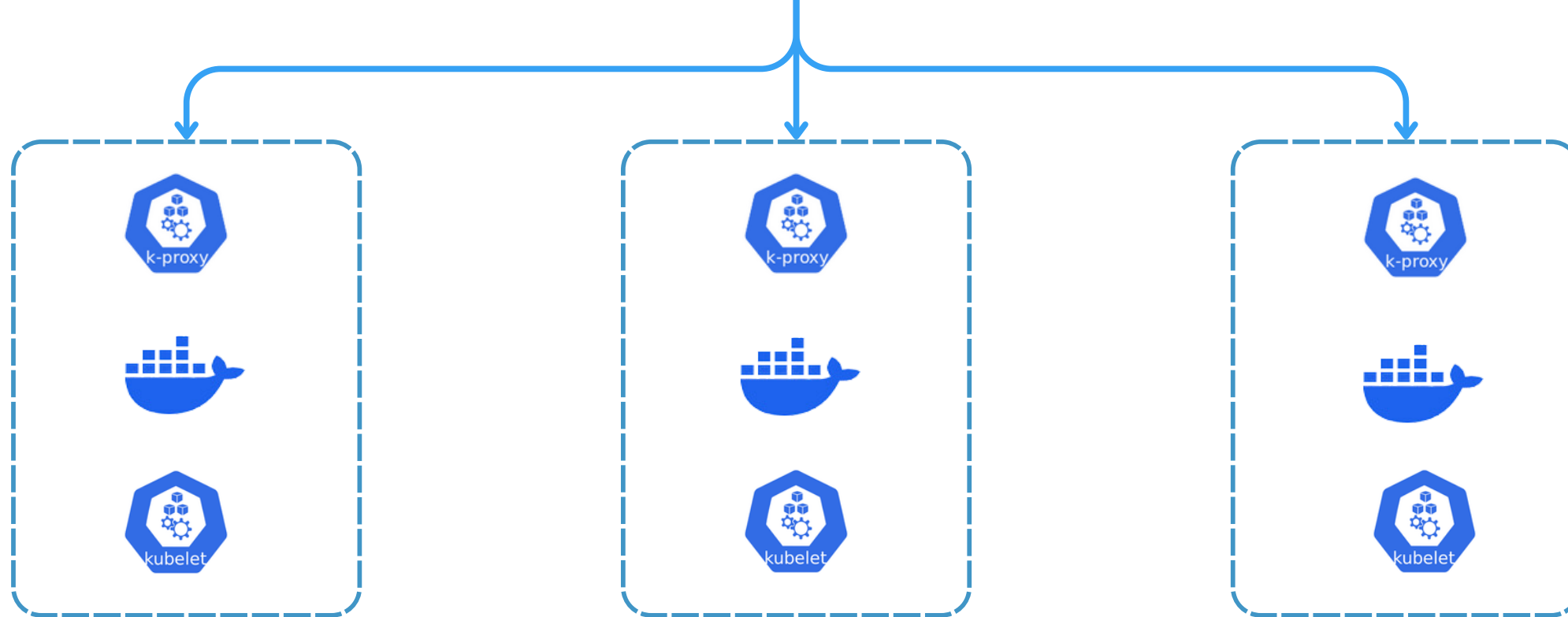


Kubernetes Basic Architecture

K8s Cluster

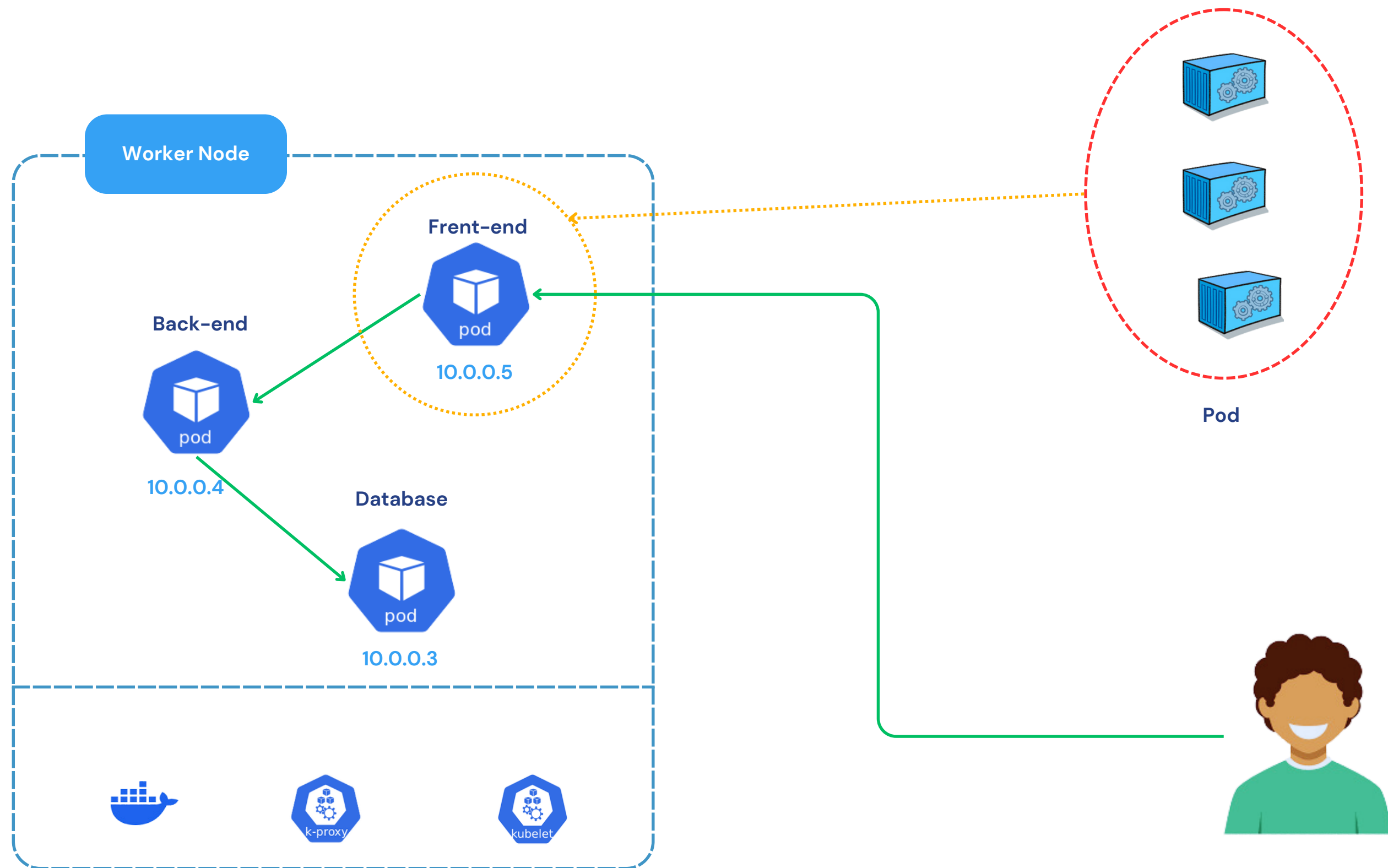


- kube-apiserver
- kube-controller-manager
- kube-scheduler
- etcd



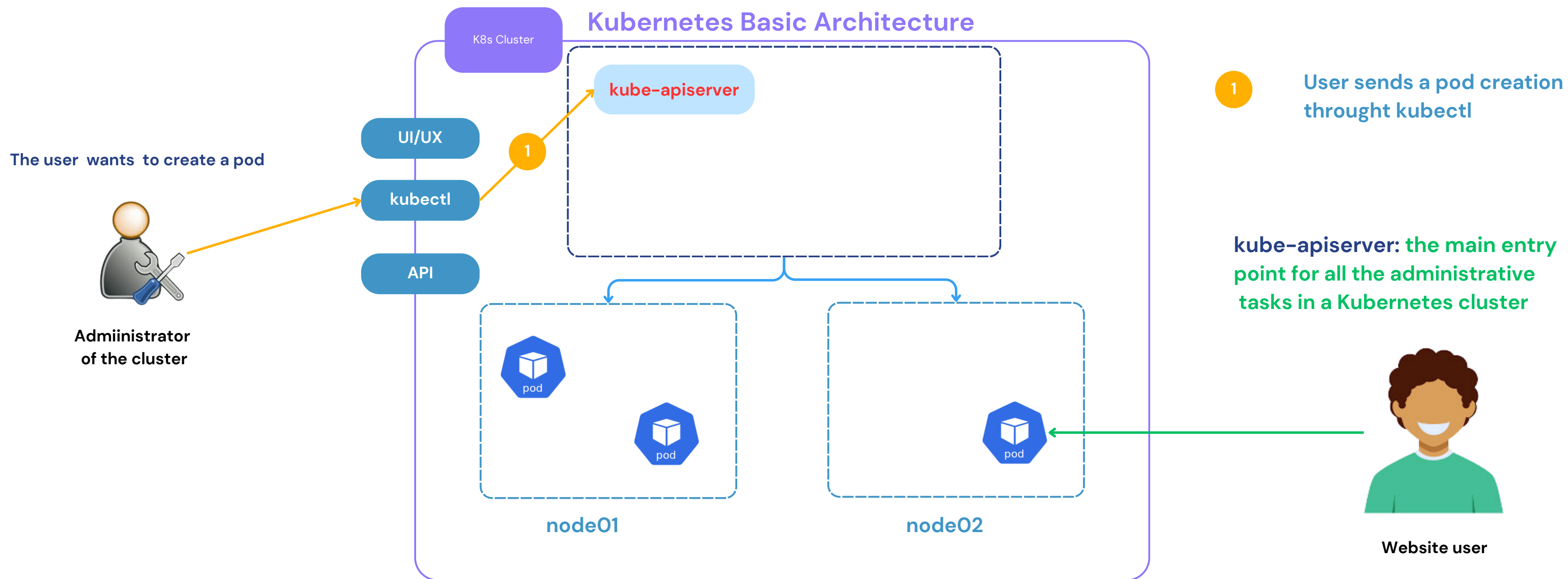
- Container runtime (Docker)
- Kubelet
- Kube-proxy

Worker nodes host Web applications, databases ...

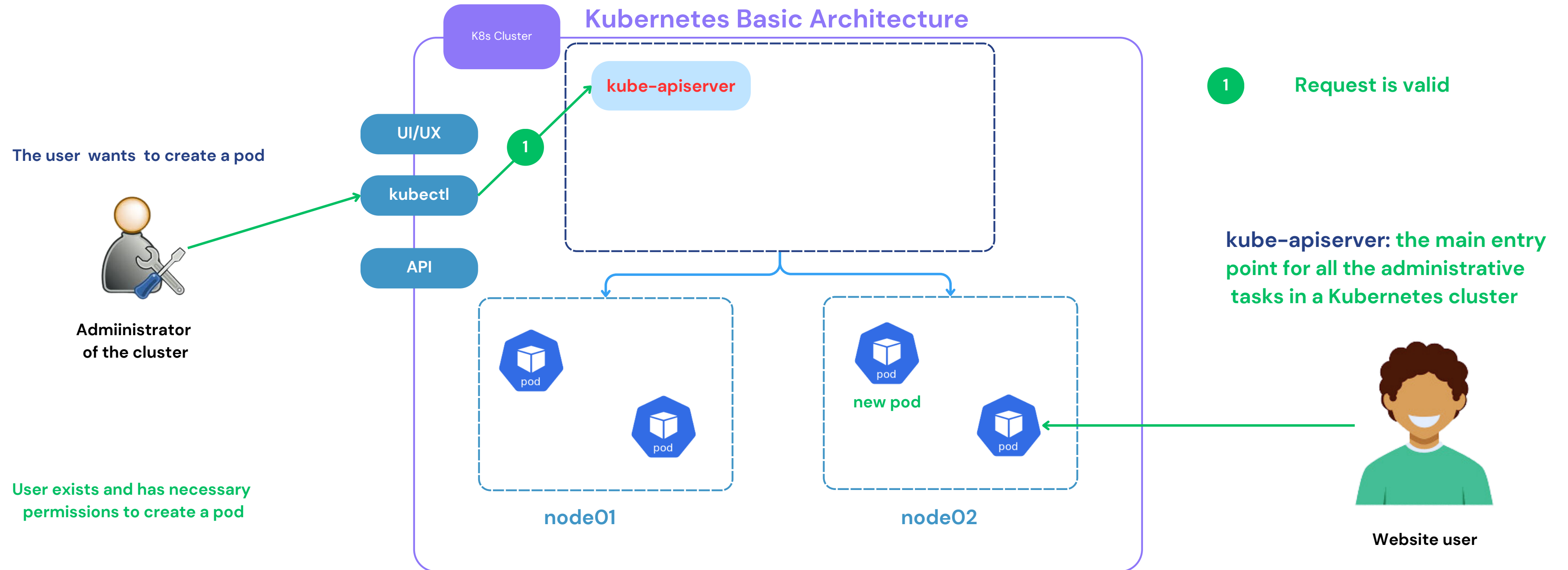


- Pod is an abstraction over containers
- A pod can contain more than one container
- One main container per pod is recommended
- Pods are ephemeral : they easily can be deleted

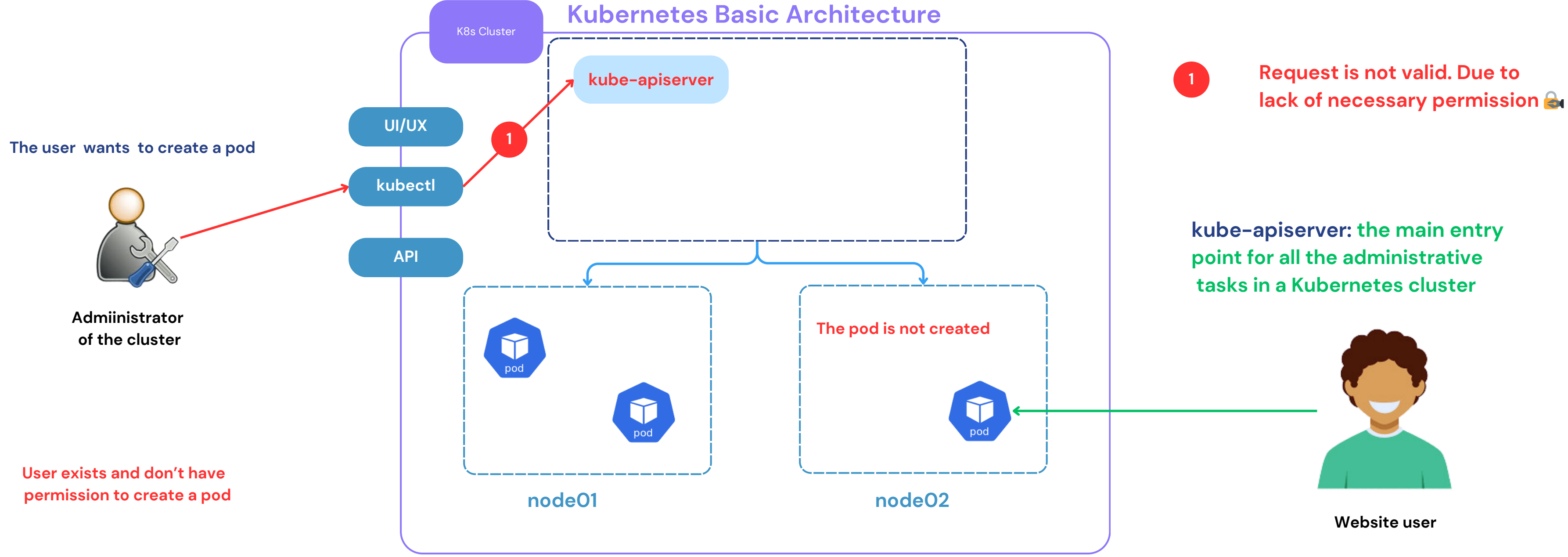
Kubernetes Basic Architecture



Kubernetes Basic Architecture

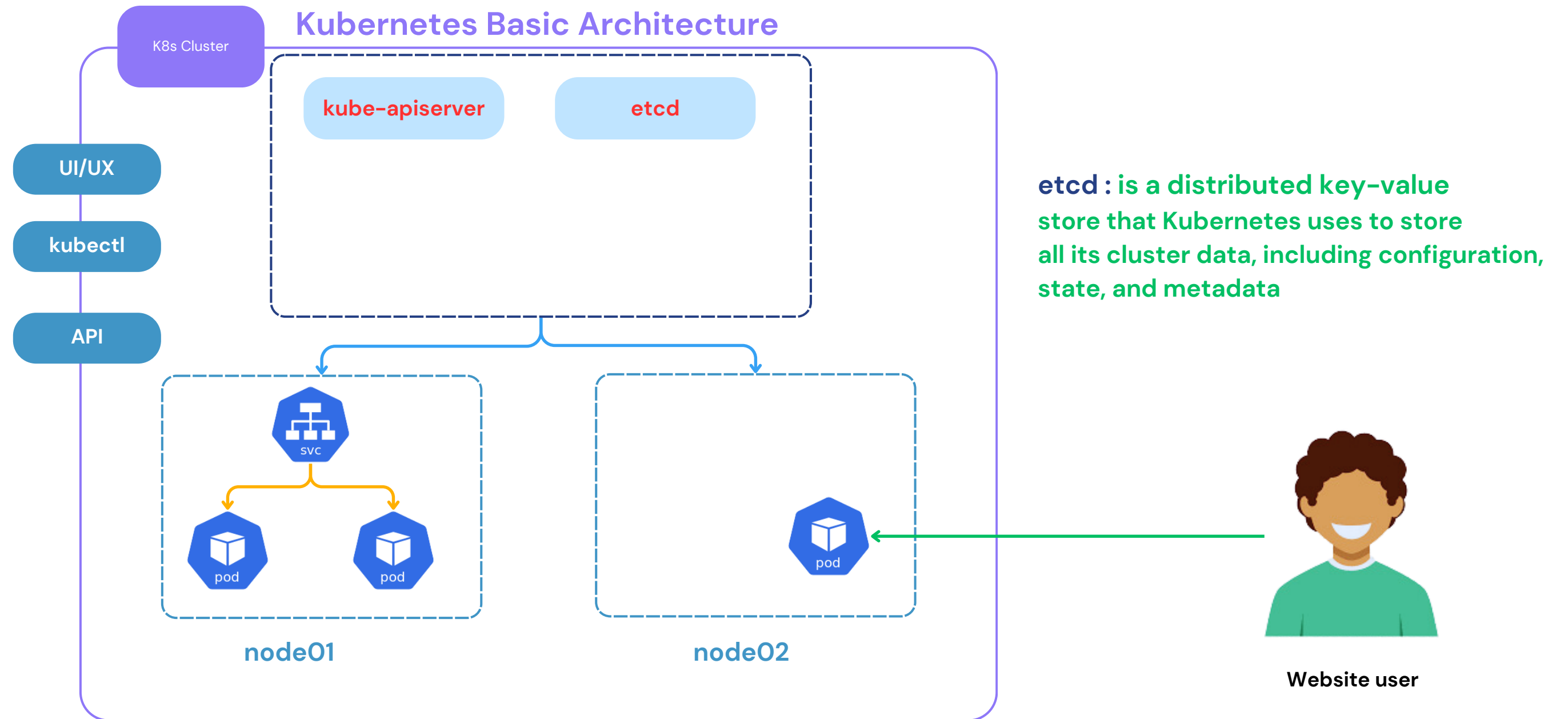


Kubernetes Basic Architecture



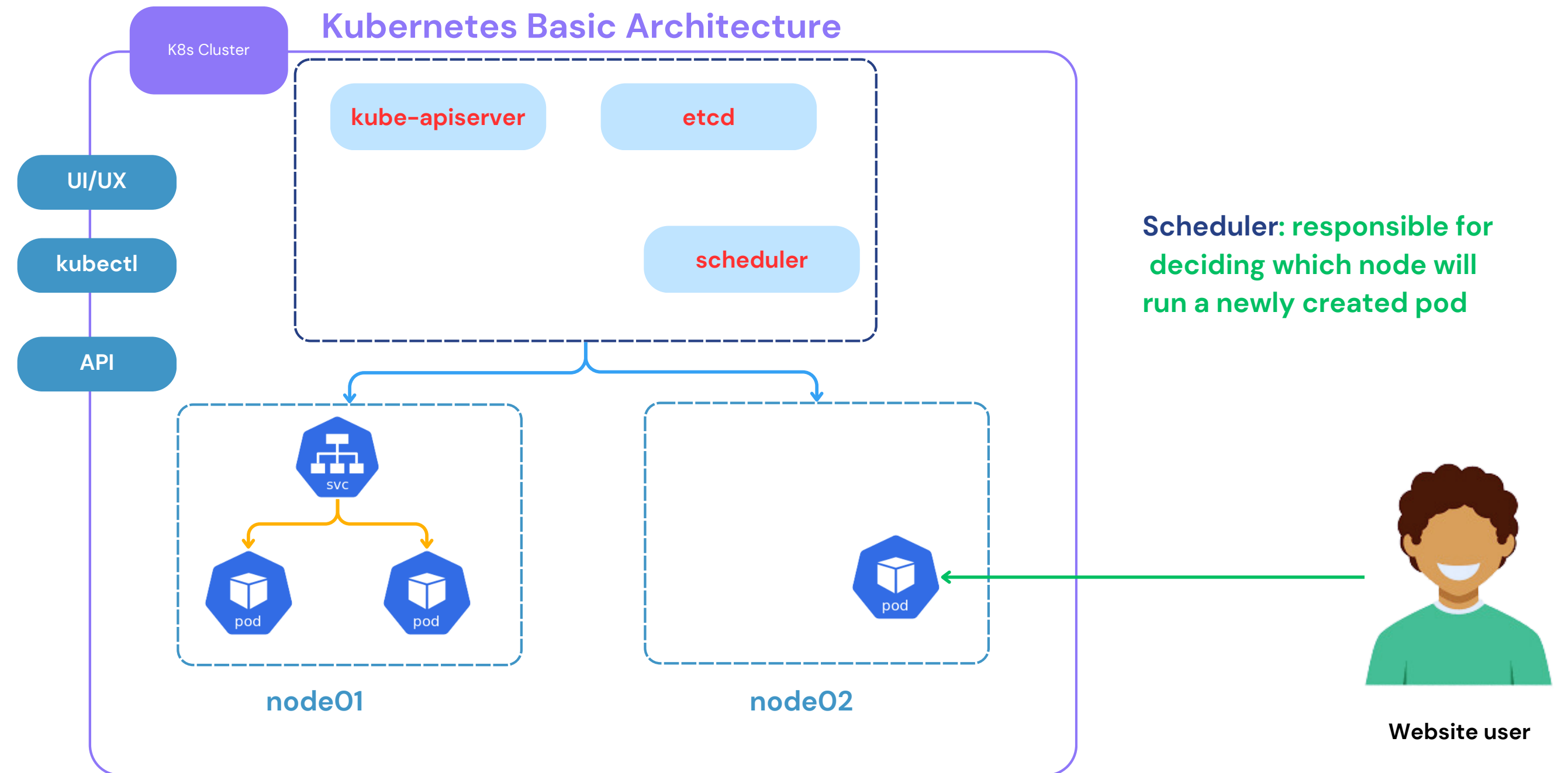
simple configuration to create a pod

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
```



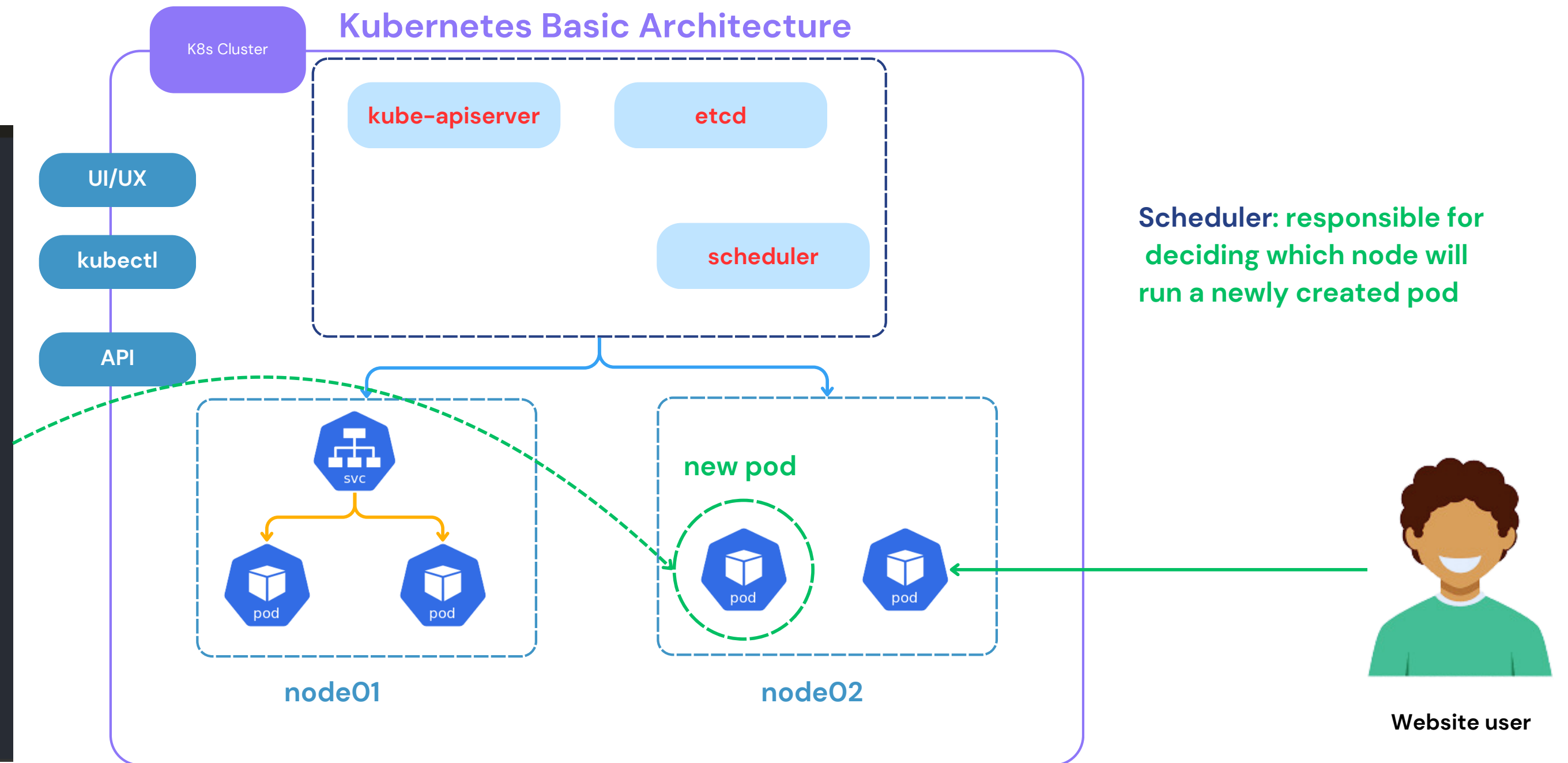
simple configuration to create a pod

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
```



simple configuration to create a pod

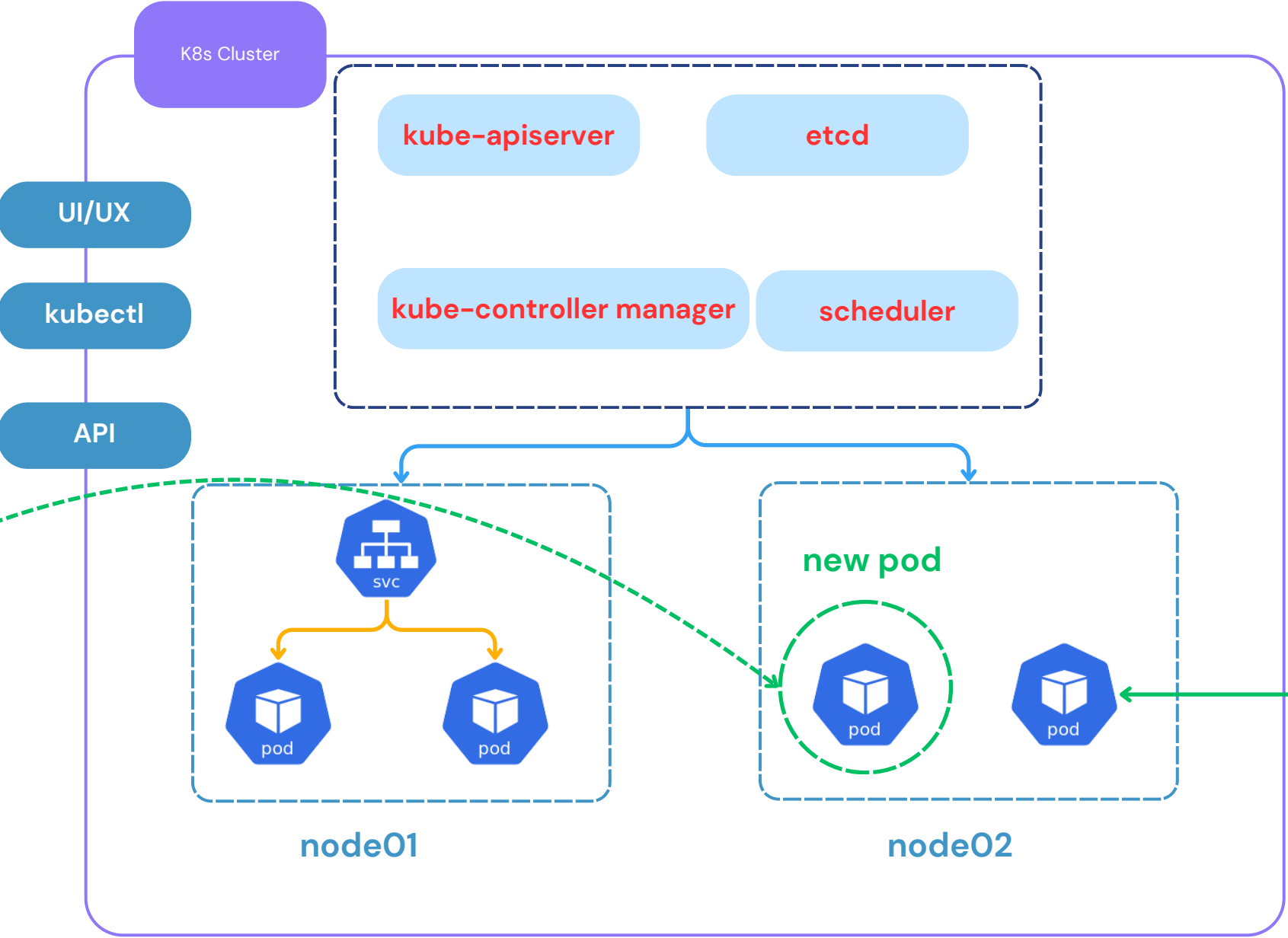
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
```



Kubernetes Basic Architecture

simple configuration of an existing pod

```
yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
status:
  phase: Running
  podIP: 10.244.0.15
  startTime: "2024-07-18T10:30:00Z"
  containerStatuses:
  - name: nginx-container
```



Controller manager : The controller manager runs controllers, which are background processes that ensure the desired state of the cluster matches the actual state. For example, if a pod goes down, a controller will ensure it gets restarted



Website user

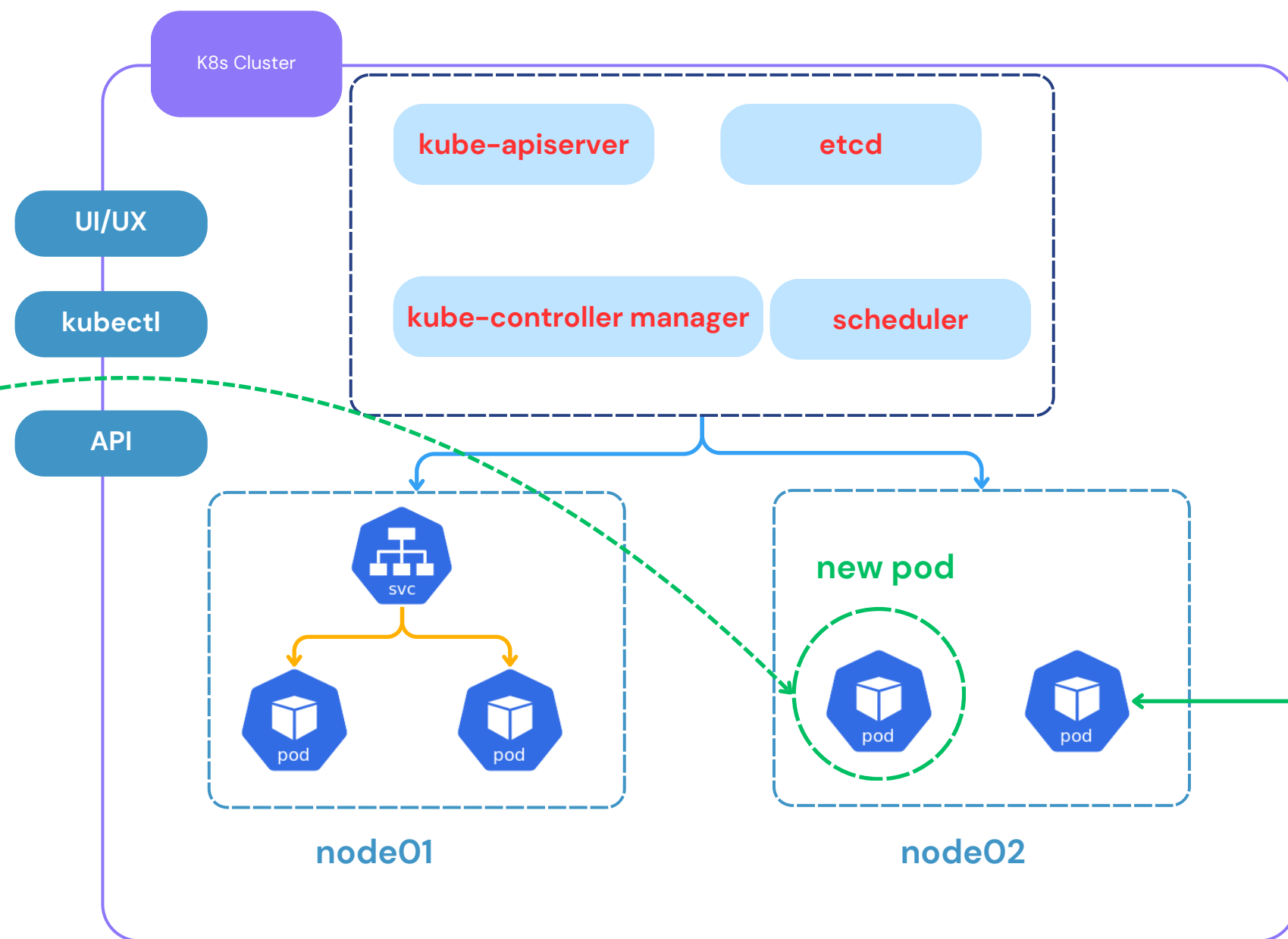
simple configuration of an existing pod

```
yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    app: nginx
spec:
  containers:
  - name: nginx-container
    image: nginx:latest
    ports:
    - containerPort: 80
status:
  phase: Running
  podIP: 10.244.0.15
  startTime: "2024-07-18T10:30:00Z"
  containerStatuses:
  - name: nginx-container
```

----- desired state

----- current state

Kubernetes Basic Architecture

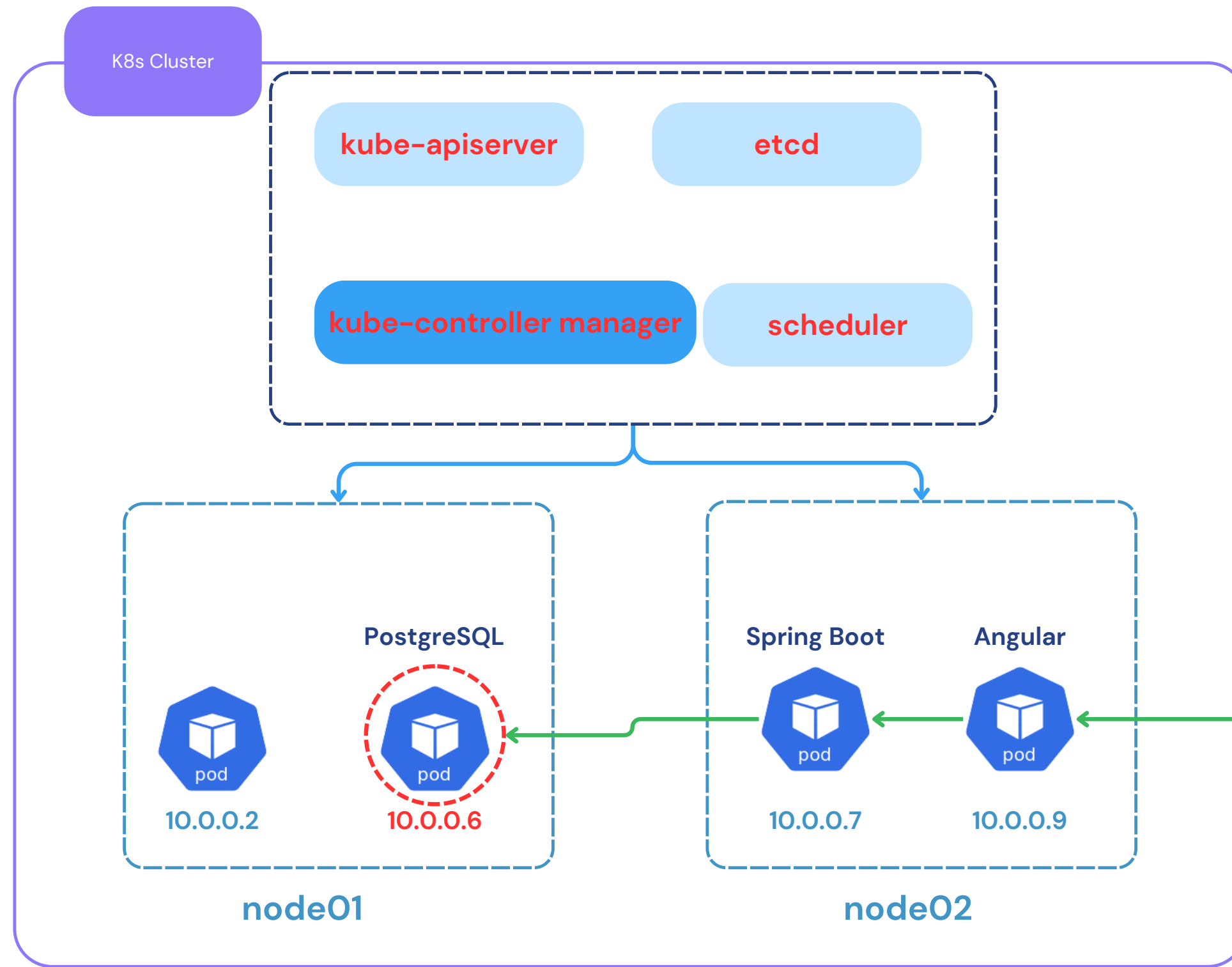


Controller manager : The controller manager runs controllers, which are background processes that ensure the desired state of the cluster matches the actual state. For example, if a pod goes down, a controller will ensure it gets restarted



Website user

Kubernetes Service

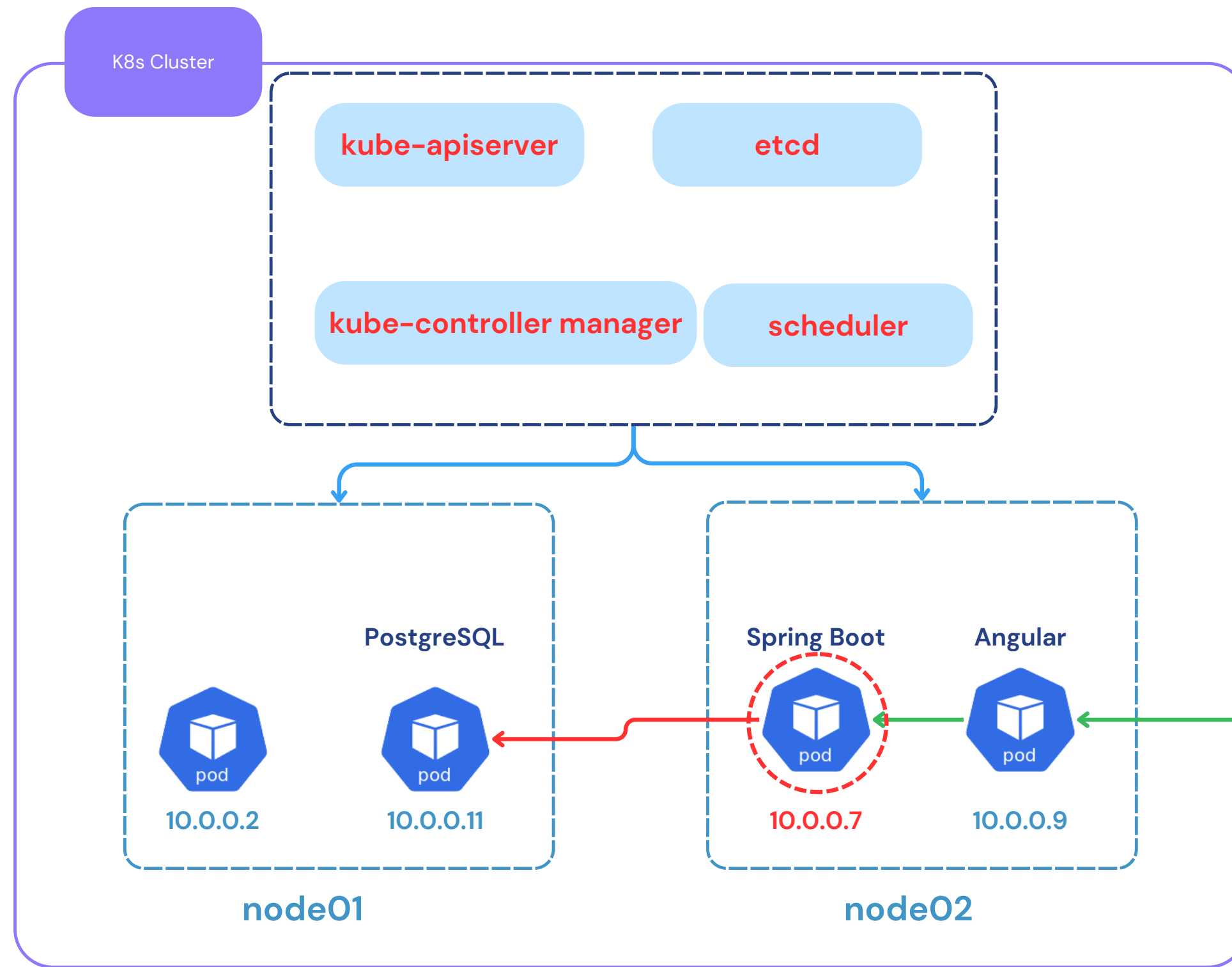


Pods are ephemeral: they can easily be terminated. When a pod stops running, it is replaced with a new pod that has a different IP address."



Website user

Kubernetes Service

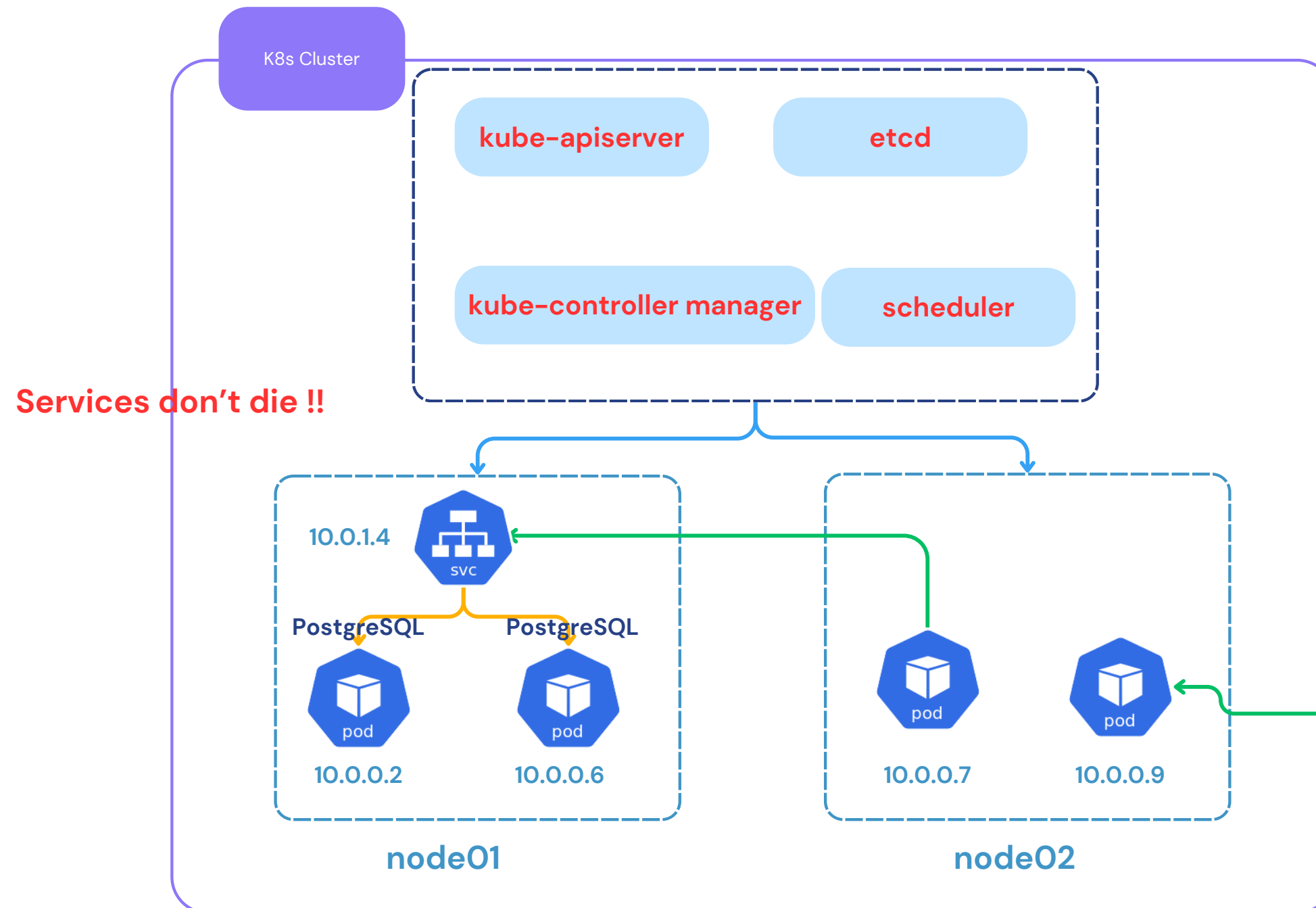


Pods are ephemeral: they can easily be terminated.
When a pod stops running, it is replaced with a new pod that has a different IP address."



Website user

Kubernetes Service



Pods are ephemeral: they can easily be terminated. When a pod stops running, it is replaced with a new pod that has a different IP address."

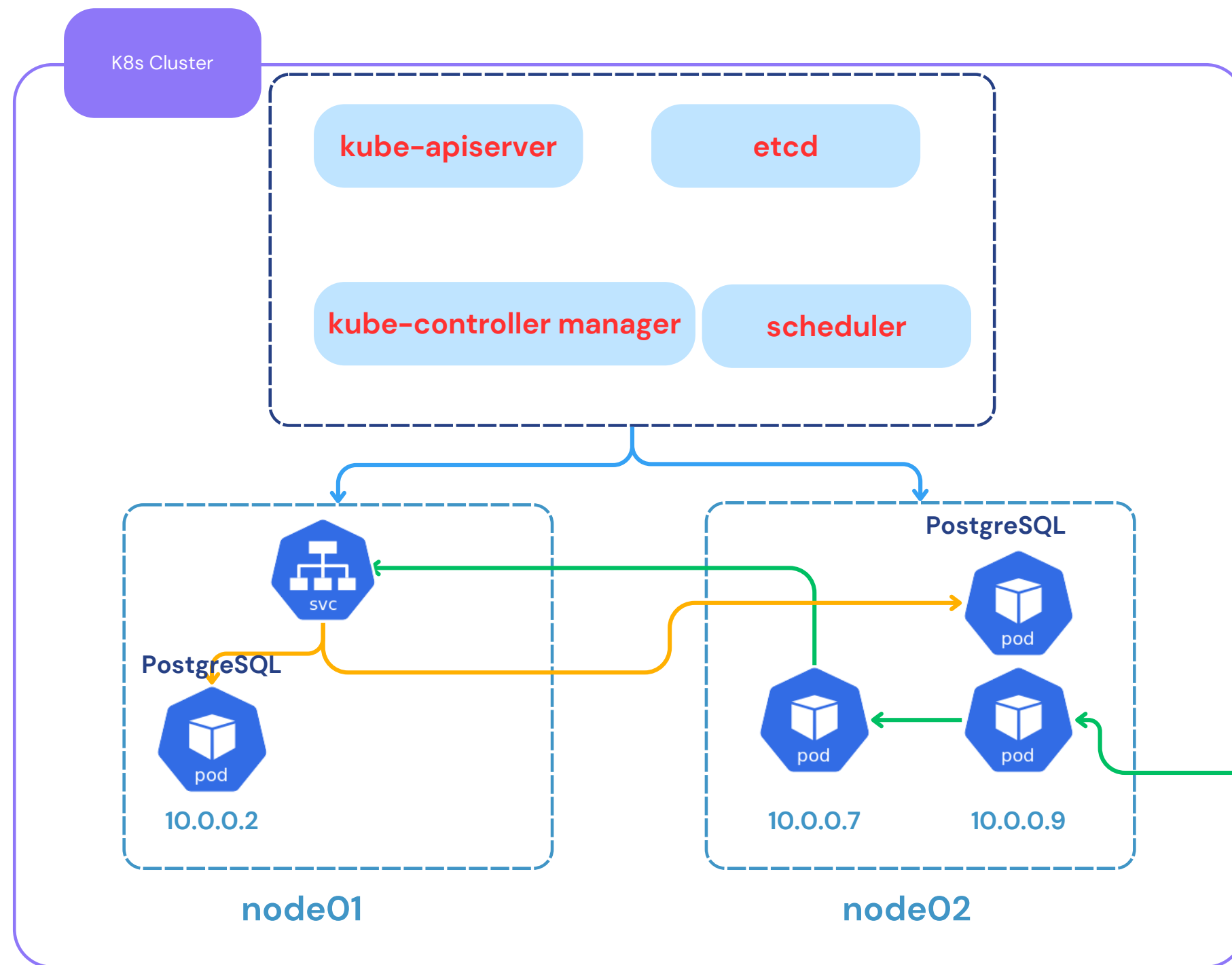
Service (svc) : provides a stable IP and DNS name for accessing a dynamic set of pods, enabling load balancing and service discovery.

```
kubectl create service clusterip db-service \
--tcp=5432:5432
```

Services Type :

- ClusterIP
- NodePort
- LoadBalancer

Kubernetes Service



Pods are ephemeral: they can easily be terminated. When a pod stops running, it is replaced with a new pod that has a different IP address."

Service (svc) : provides a stable IP and DNS name for accessing a dynamic set of pods, enabling load balancing and service discovery.

```
kubectl create service clusterip db-service \
--tcp=5432:5432
```

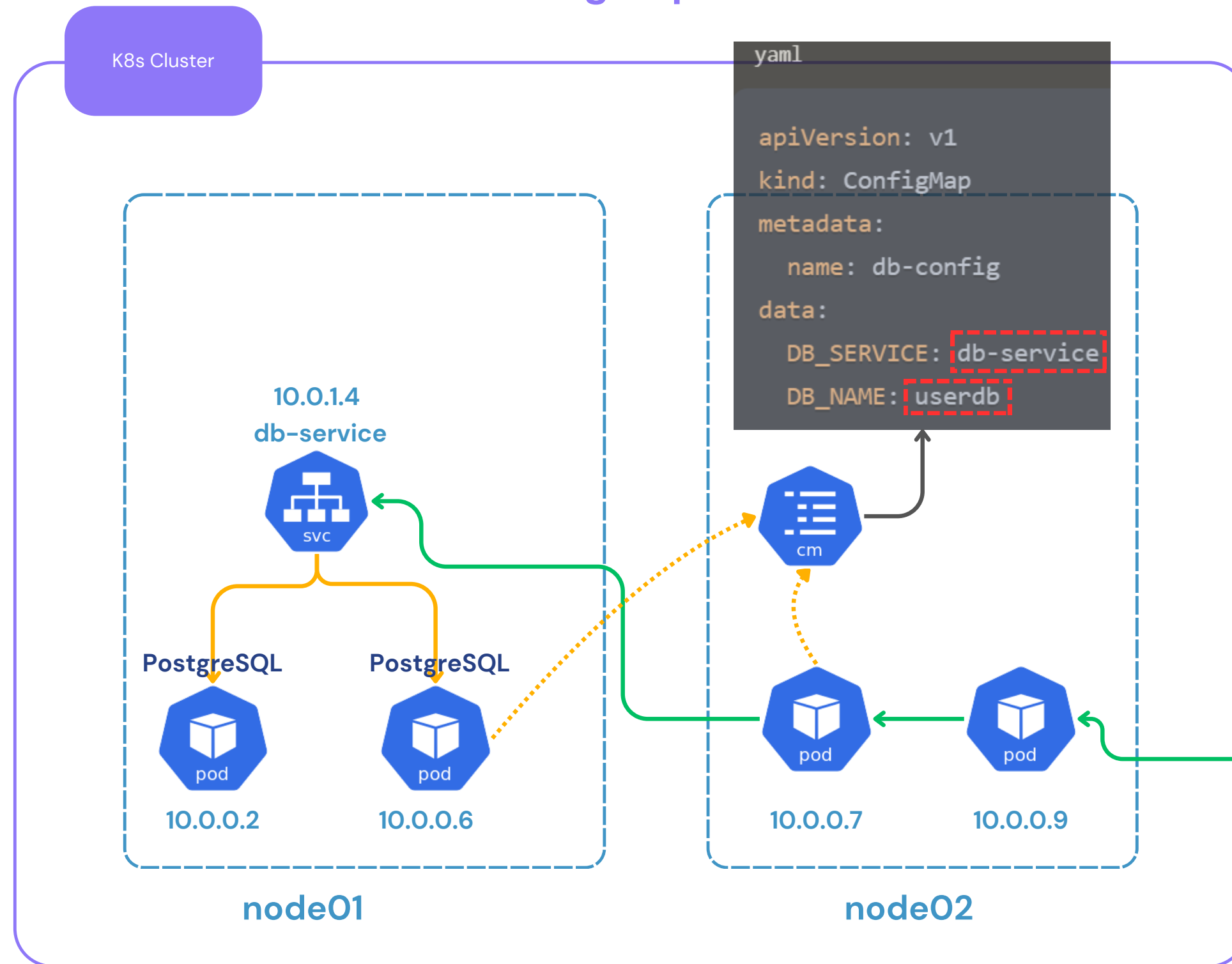


Website user

Services Type :

- ClusterIP
- NodePort
- LoadBalancer

ConfigMap



ConfigMap in Kubernetes : stores key-value pairs, allowing applications to retrieve configuration data at runtime without using hardcoded values in the code.

```
properties

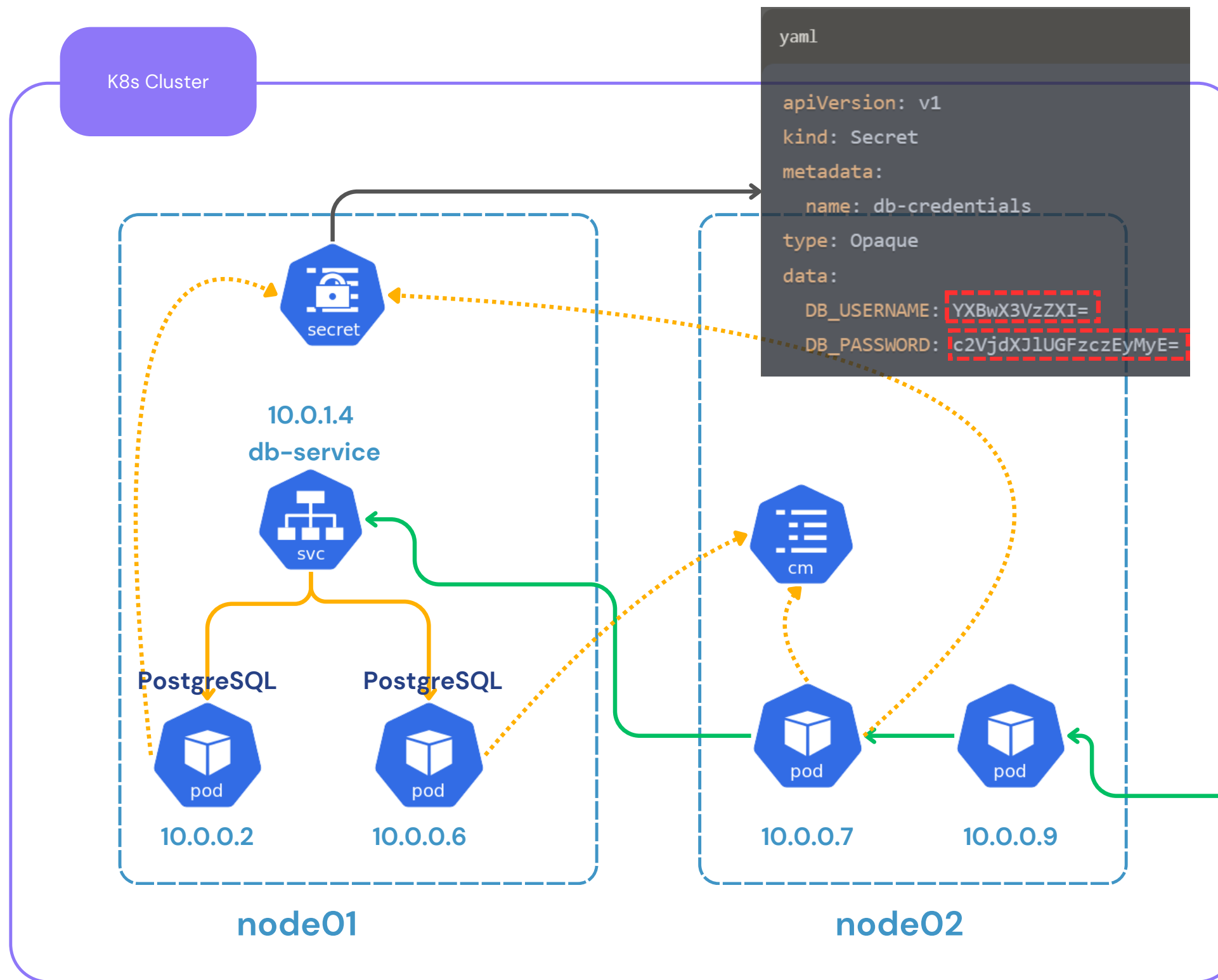
spring.datasource.url=jdbc:postgresql://db-service:5432/userdb
spring.datasource.username=app_user
spring.datasource.password=securePass123!
```

```
kubectl create cm db-service \
  --from-literal=DB_SERVICE=db-service \
  --from-literal=DB_NAME=userdb
```



Website user

Secret



Secret: Kubernetes object that securely stores and manages sensitive information, such as passwords, OAuth tokens, and SSH keys.

```
properties
spring.datasource.url=jdbc:postgresql://db-service:5432/userdb
spring.datasource.username=app_user
spring.datasource.password=securePass123!
```

```
kubectl create secret generic db-credentials \
--from-literal=DB_USERNAME=app_user \
--from-literal=DB_PASSWORD=securePass123!
```

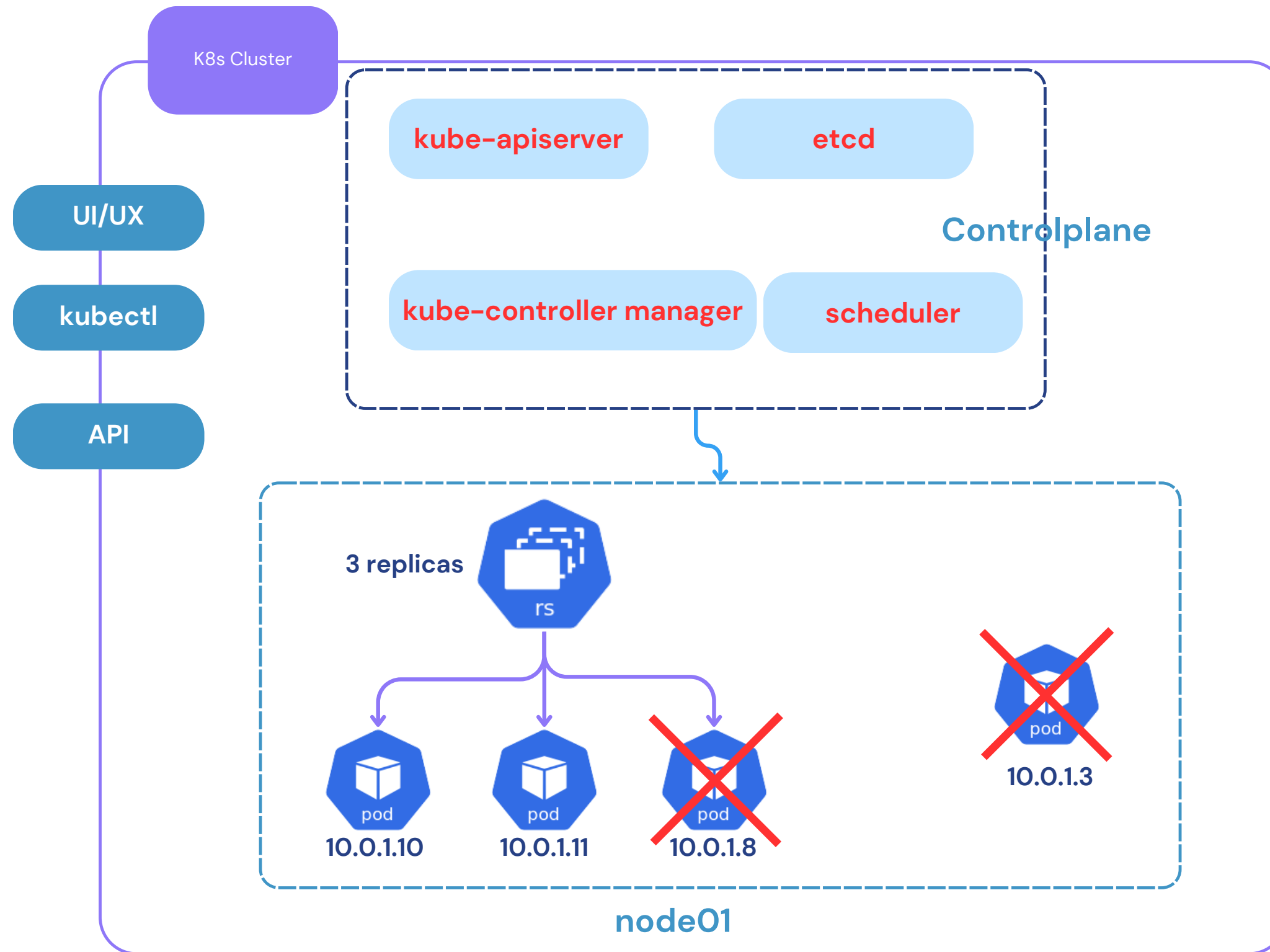


Website user

Secret types :

- Opaque (generic)
- kubernetes.io/ssh-auth
- kubernetes.io/tls
- kubernetes.io/service-account-token
-

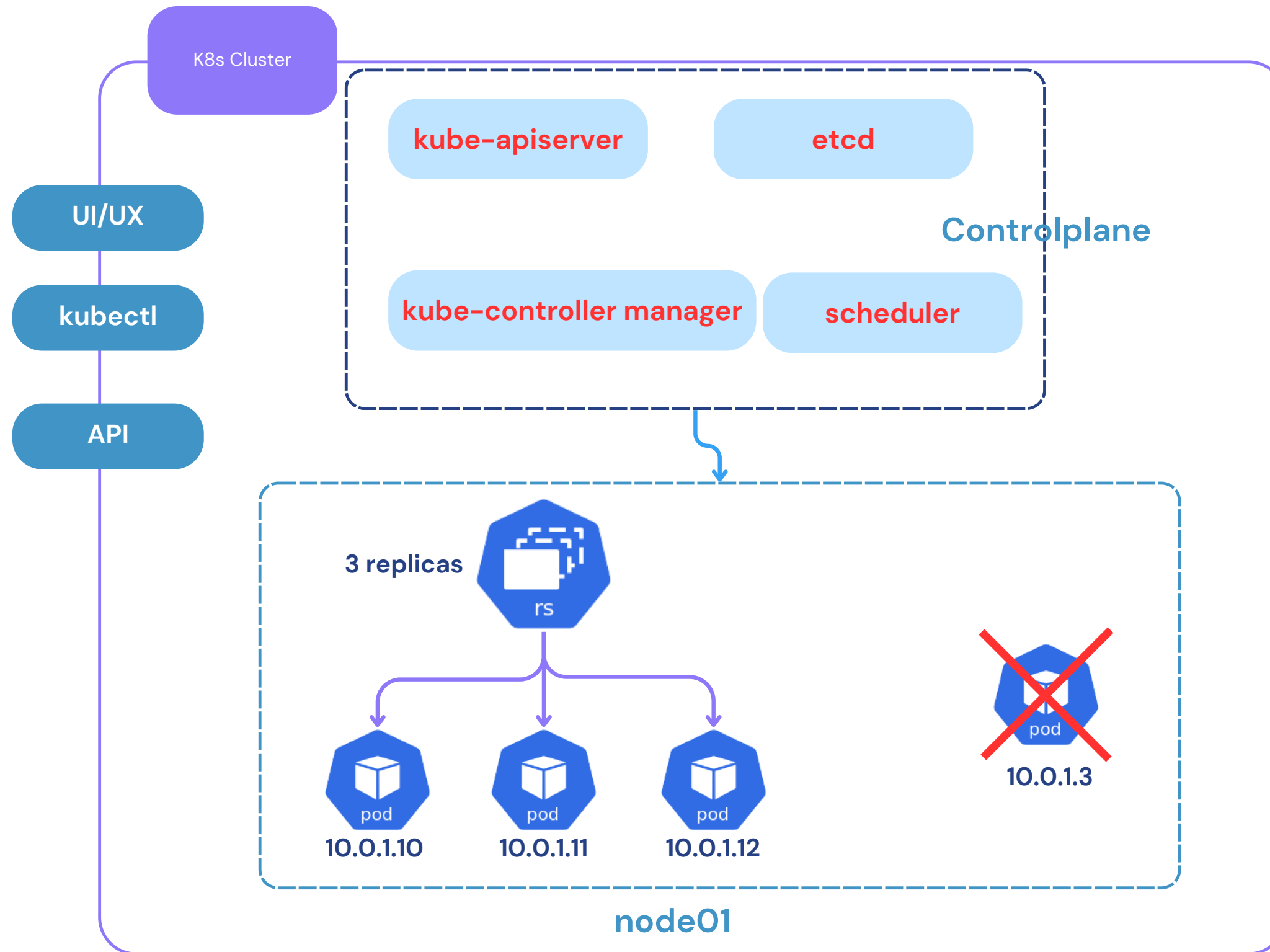
Replicaset



replicaset : is a mechanism that ensures a specified number of identical pods are running at any given time.

The **controller manager** focuses on maintaining desired state for replica sets, not individual pods

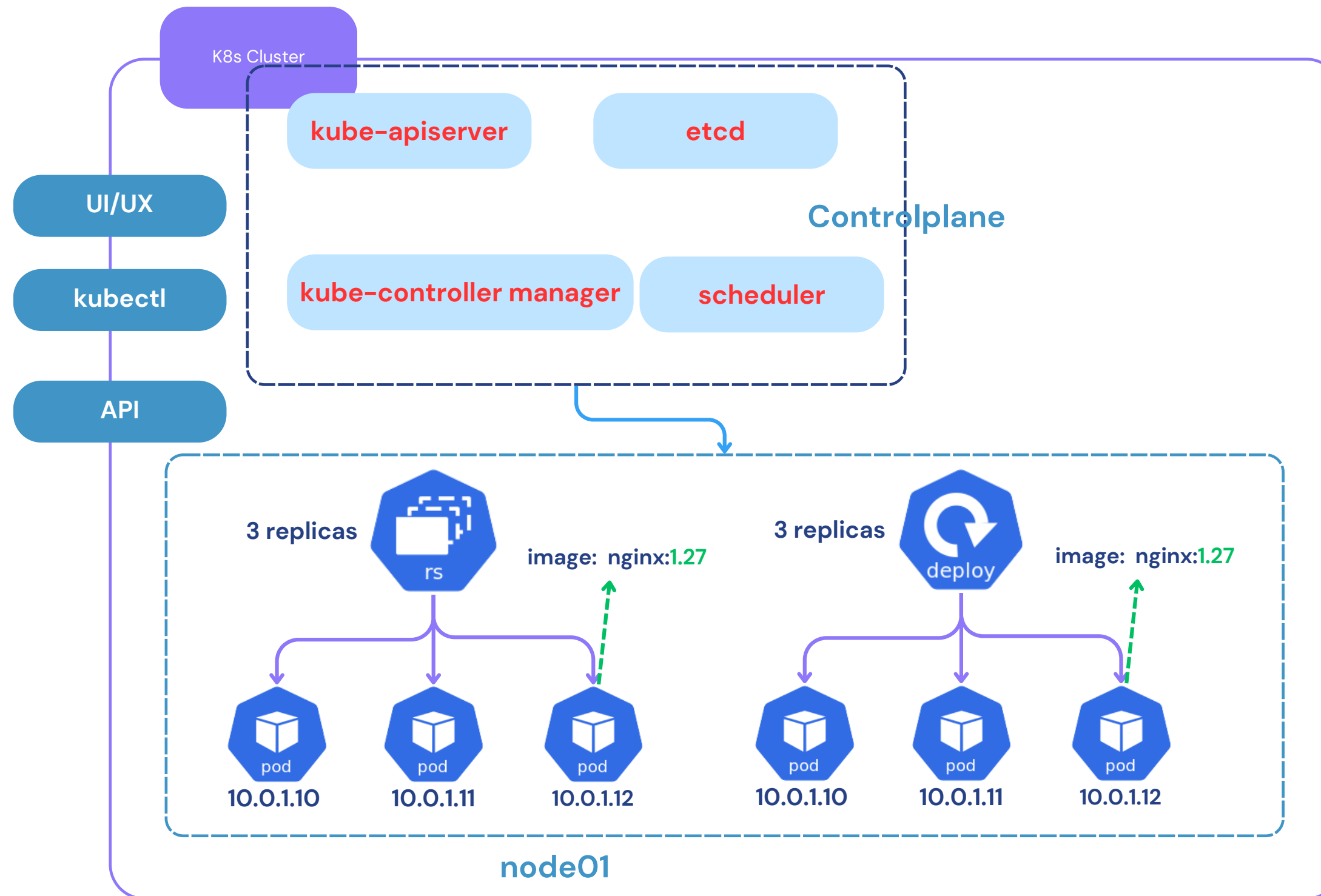
Replicaset



replicaset : is a mechanism that ensures a specified number of identical pods are running at any given time.

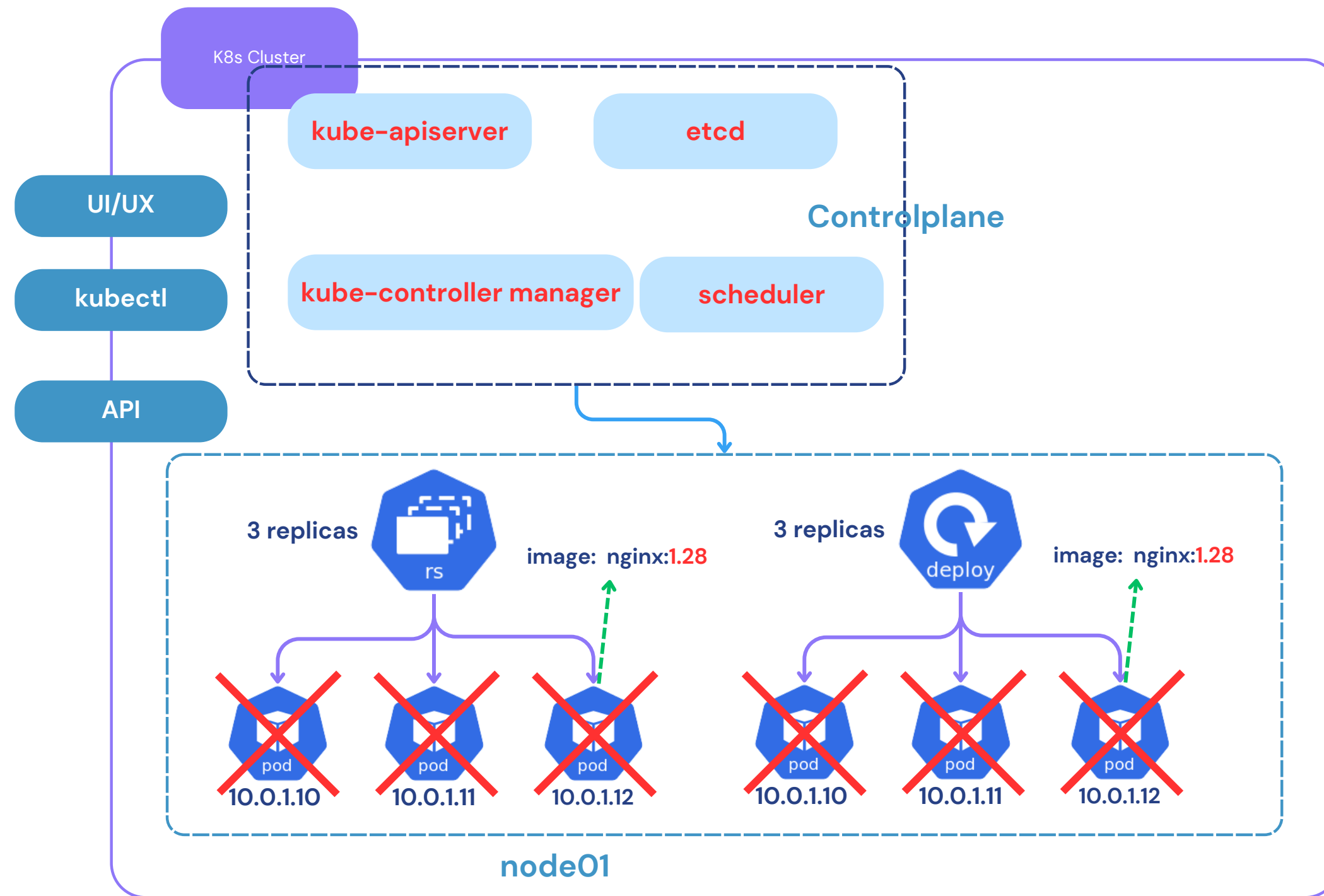
The **controller manager** focuses on maintaining desired state for replica sets, not individual pods

Deployment



If updating a pod's version **fails**, you would need to **manually** remove the **replicaset** and create a new one with the **previous version** until the issue is resolved.

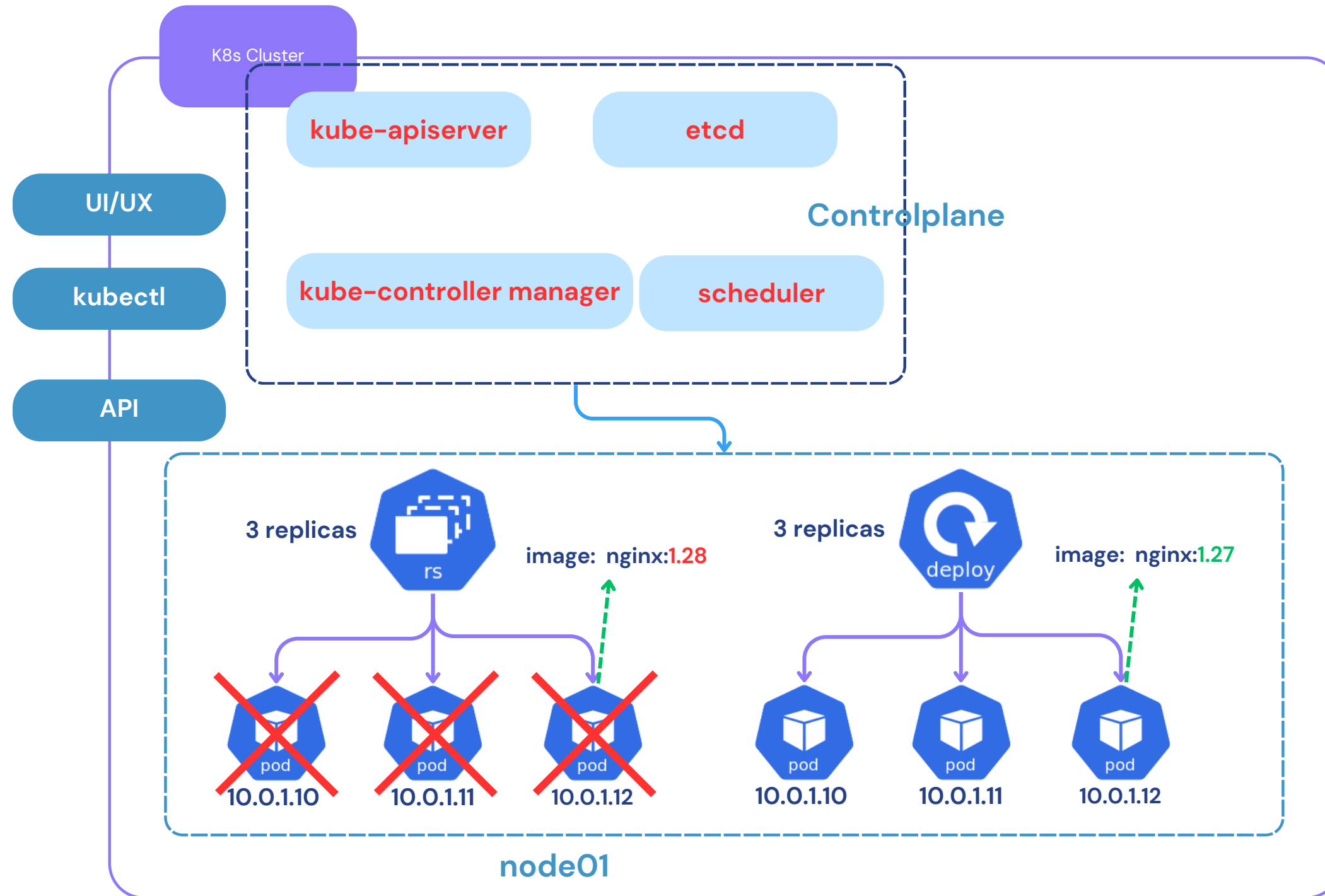
Deployment



If updating a pod's version **fails**, you would need to **manually** remove the replicaset and create a new one with the previous version until the issue is resolved.

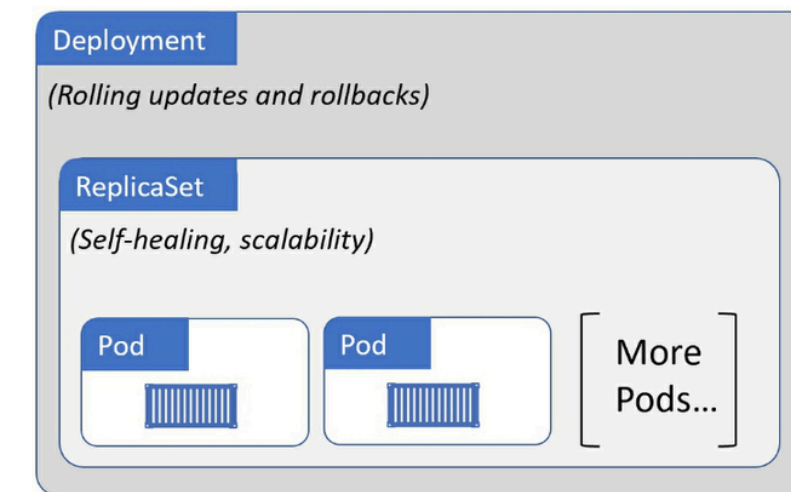
Deployment : is a resource that automates and simplifies the update process for applications. Unlike managing ReplicaSets directly, Deployments handle version changes and rollbacks automatically

Deployment

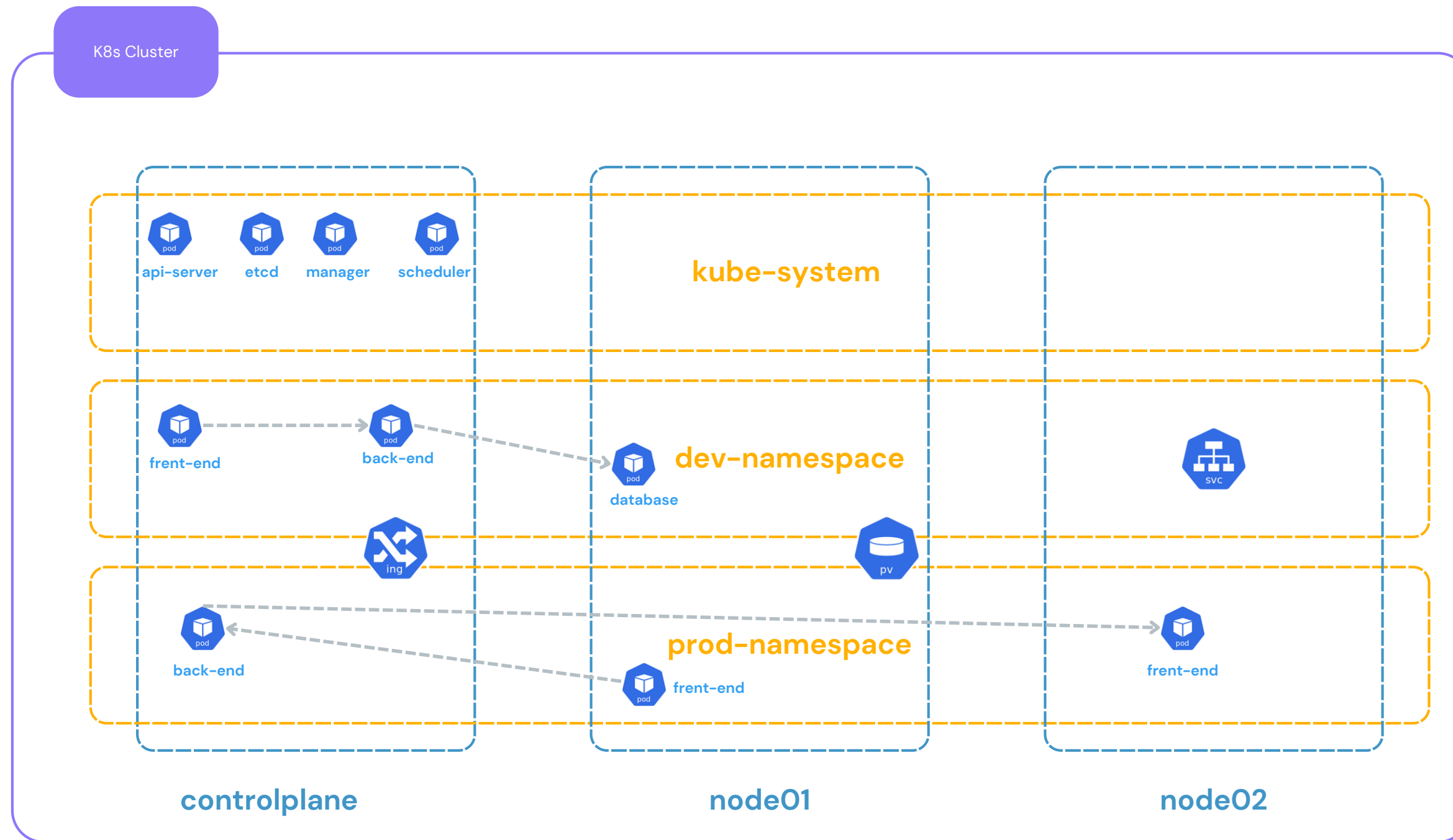


If updating a pod's version **fails**, you would need to **manually** remove the **replicaset** and create a new one with the **previous version** until the issue is resolved.

Deployment : is a resource that automates and simplifies the update process for applications. Unlike managing ReplicaSets directly, Deployments handle version changes and rollbacks automatically



Namespace



Namespace : is a logical partition that allows you to divide cluster resources among multiple users or applications, providing a scope for names and isolating resources within the same cluster.

```
kubectl get namespaces
```

```
kubectl get pods -n kube-system
```

```
kubectl api-resources --namespaced
```

Breaking down the structure of Kubernetes YAML files

yaml

apiVersion: v1

kind: Pod

metadata:

name: my-pod

labels:

app: my-app

spec:

containers:

- name: my-container

image: nginx:latest

ports:

- containerPort: 80

status:

phase: Running

podIP: 10.244.0.5

startTime: "2024-07-20T12:00:00Z"

conditions:

- type: Ready

yaml

apiVersion: v1

kind: Service

metadata:

name: my-service

namespace: my-namespace

spec:

selector:

app: my-app

ports:

- protocol: TCP

port: 80

targetPort: 8080

type: ClusterIP

status:

loadBalancer: {}

yaml

apiVersion: apps/v1

kind: Deployment

metadata:

name: my-deployment

namespace: my-namespace

spec:

replicas: 3

selector:

matchLabels:

app: my-app

template:

metadata:

labels:

app: my-app

spec:

containers:

- name: my-container

image: nginx:latest

ports:

- containerPort: 8080

status:

observedGeneration: 1

Thank you for your attention!

For more insights and updates,
connect with me on LinkedIn



[Redouane Soul](#)