Kubernetes
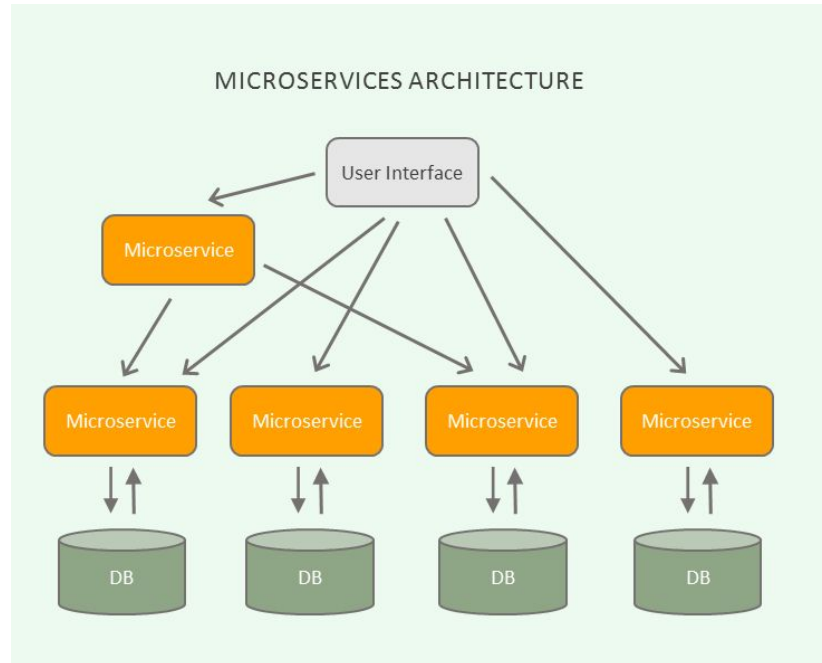
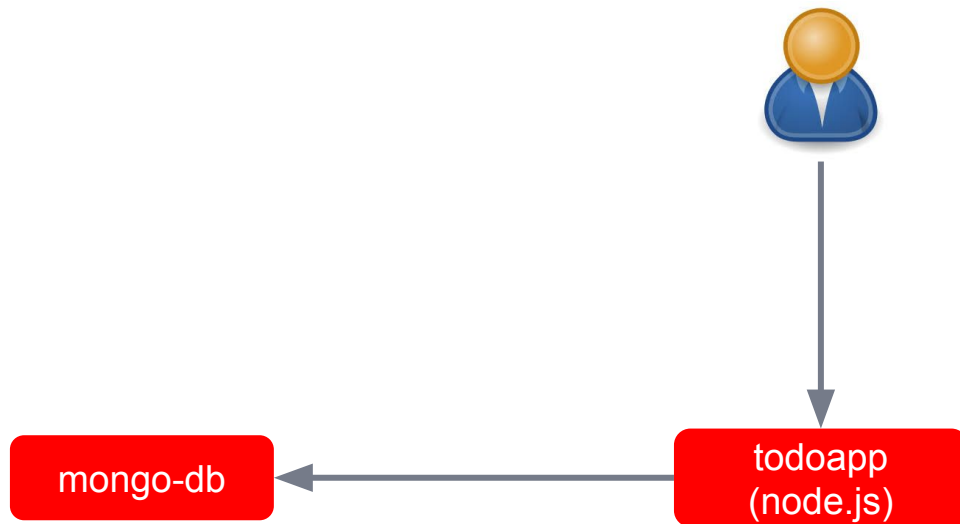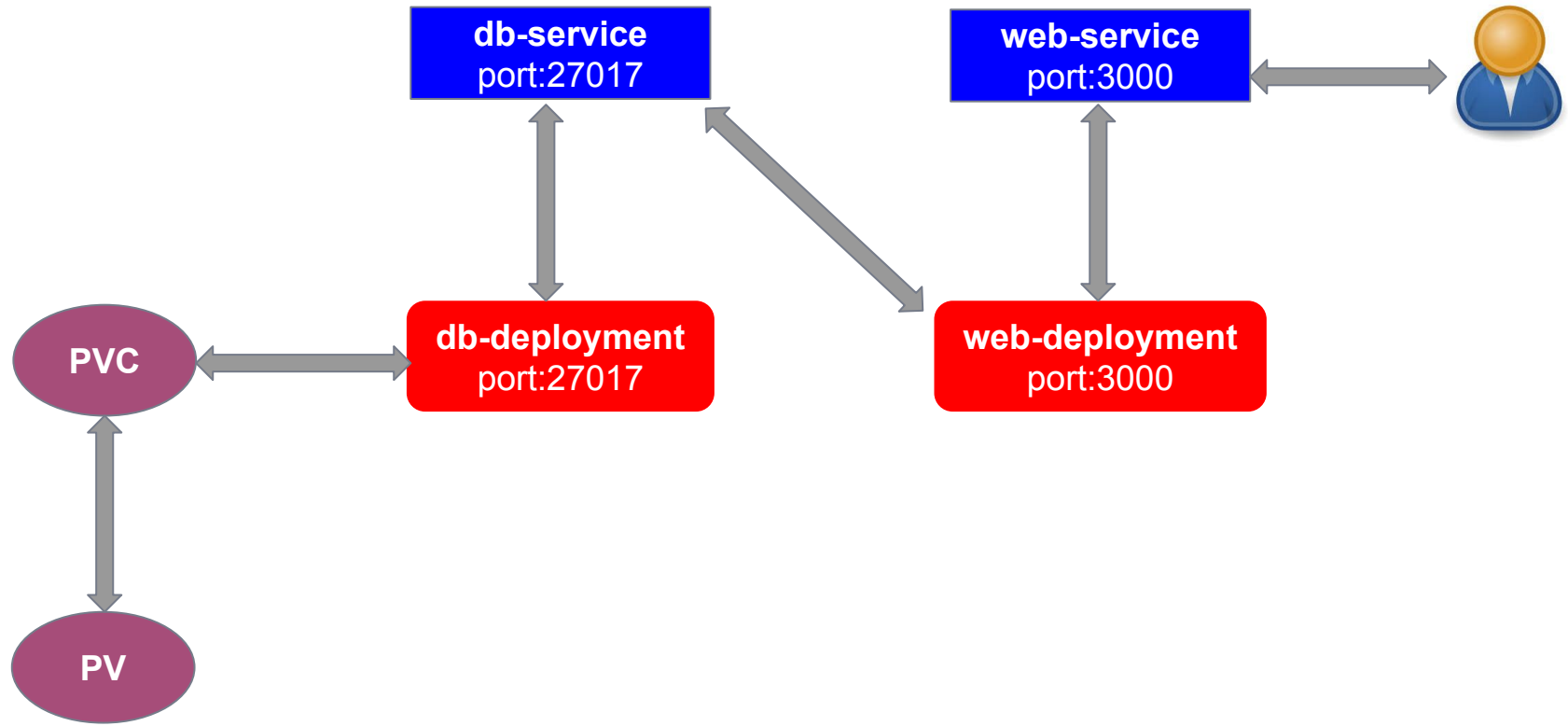# Microservices

A **microservices architecture** breaks it down into a collection of smaller independent units.

MICROSERVICES ARCHITECTURE

Microservices

mongo-db

todoapp
(node.js)

Microservices

CLARUSWAY©
WAY TO REINVENT YOURSELF

Labels and Label Selectors

- **Labels** are key-value pairs attached to Kubernetes objects (e.g. Pods, ReplicaSets, Nodes, Namespaces, Persistent Volumes).
- Labels are used to organize and select a subset of objects, based on the requirements in place.
- Many objects can have the same Label(s). Labels do not provide uniqueness to objects.
- Controllers use Labels to logically group together decoupled objects, rather than using objects' names or IDs.

Example labels:

- "environment" : "dev", "environment" : "qa", "environment" : "production"
- "tier" : "frontend", "tier" : "backend", "tier" : "cache"

Labels and Label Selectors

Controllers use **Label Selectors** to select a subset of objects.

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: db-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      name: mongo
  template:
    metadata:
      labels:
        name: mongo
        app: todoapp
    spec:
      containers:
```

```yaml
apiVersion: v1
kind: Service
metadata:
  name: db-service
  labels:
    name: mongo
    app: todoapp
spec:
  selector:
    name: mongo
  type: ClusterIP
  ports:
    - name: db
      port: 27017
      targetPort: 27017
```

CLARUSWAY©
WAY TO REINVENT YOURSELF

THANKS!

Any questions?

You can find me at:

▸   Joe@clarusway.com

CLARUSWAY©
WAY TO REINVENT YOURSELF