STATEFUL APPLICATIONS MIT KUBERNETES

BED-CON 2017

Nicolas Byl, codecentric AG

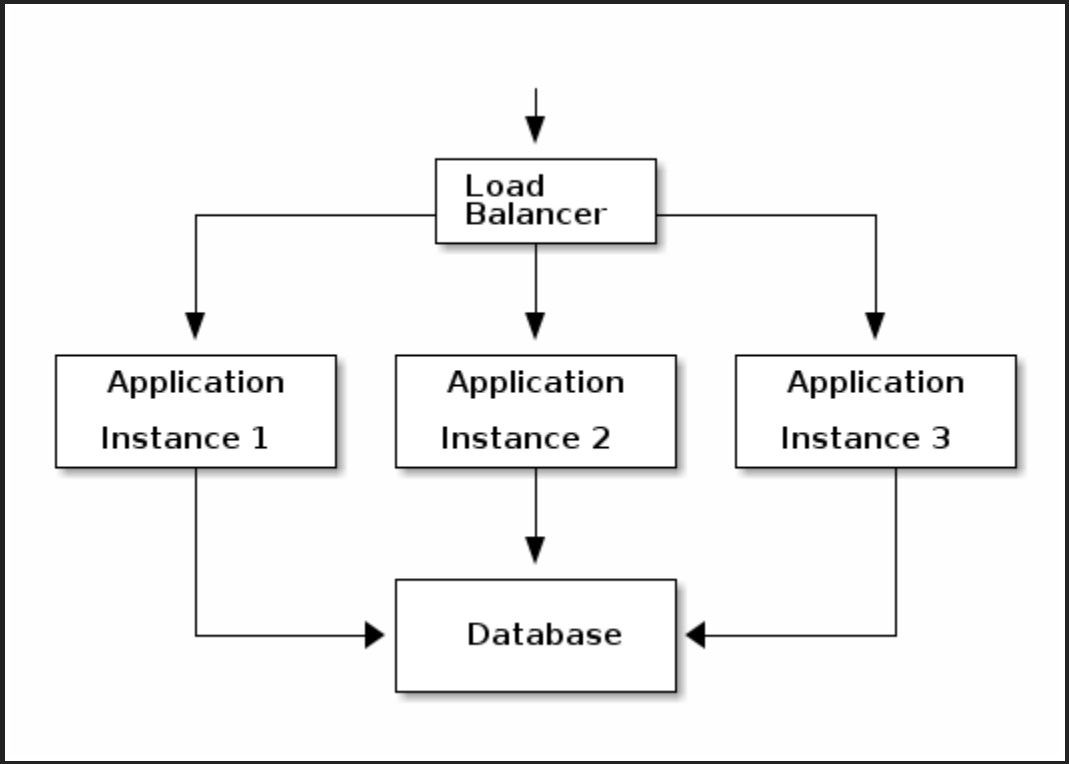

WE'RE GONNA NEED A
BIGGER DATABASE
memegenerator.net

PERSISTENT APPLICATION STATE

GOALS

- Scalable
- Highly Available
- Fault Tolerance

EXAMPLE ARCHITECTURE



CANDIDATES

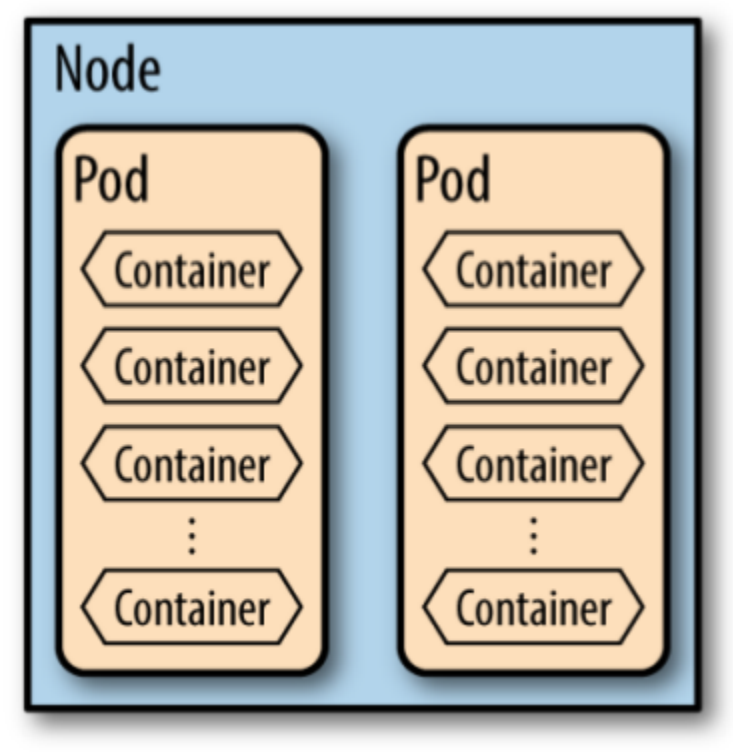
MySQL	PostgreSQL
Cassandra	MongoDB
Kafka	…

KUBERNETES

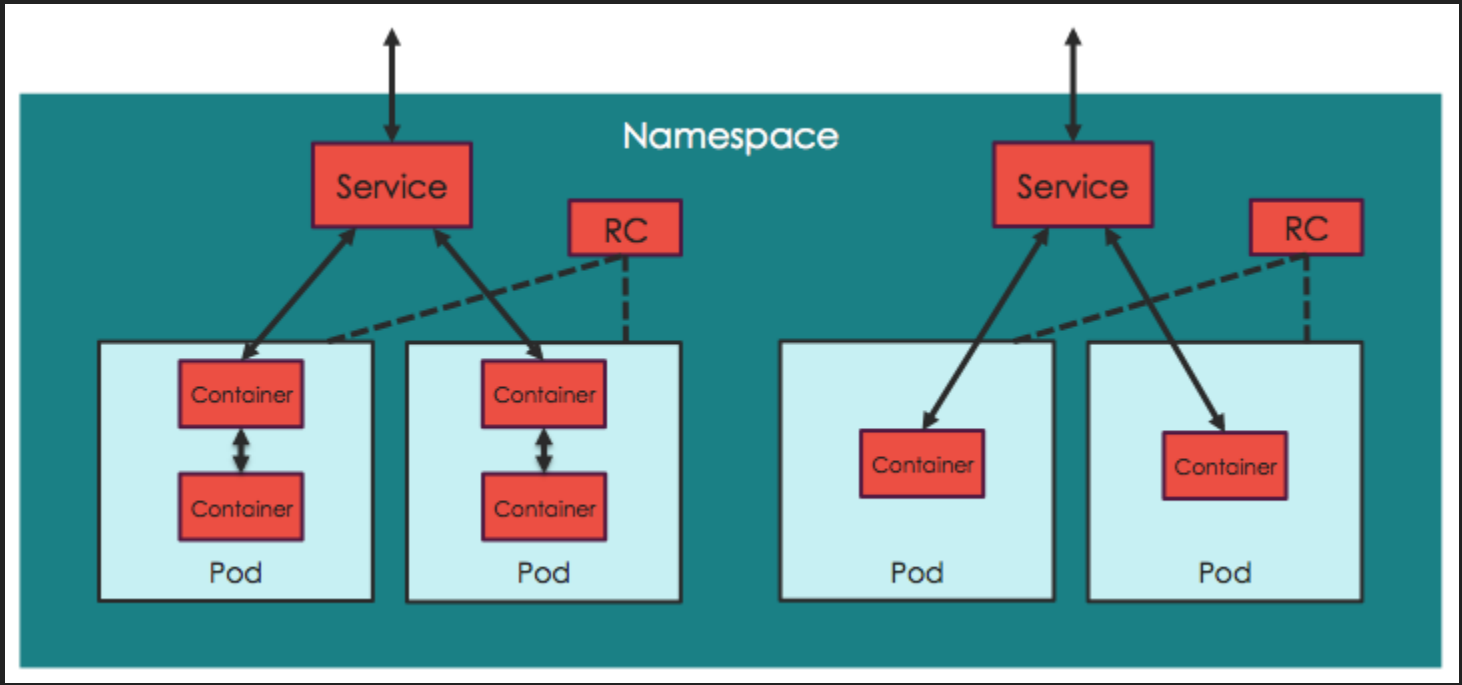
DESIGN PHILOSOPHY

- **portable**: public, private, hybrid, multi-cloud
- **extensible**: modular, pluggable, hookable, composable
- **self-healing**: auto-placement, auto-restart, auto-replication, auto-scaling

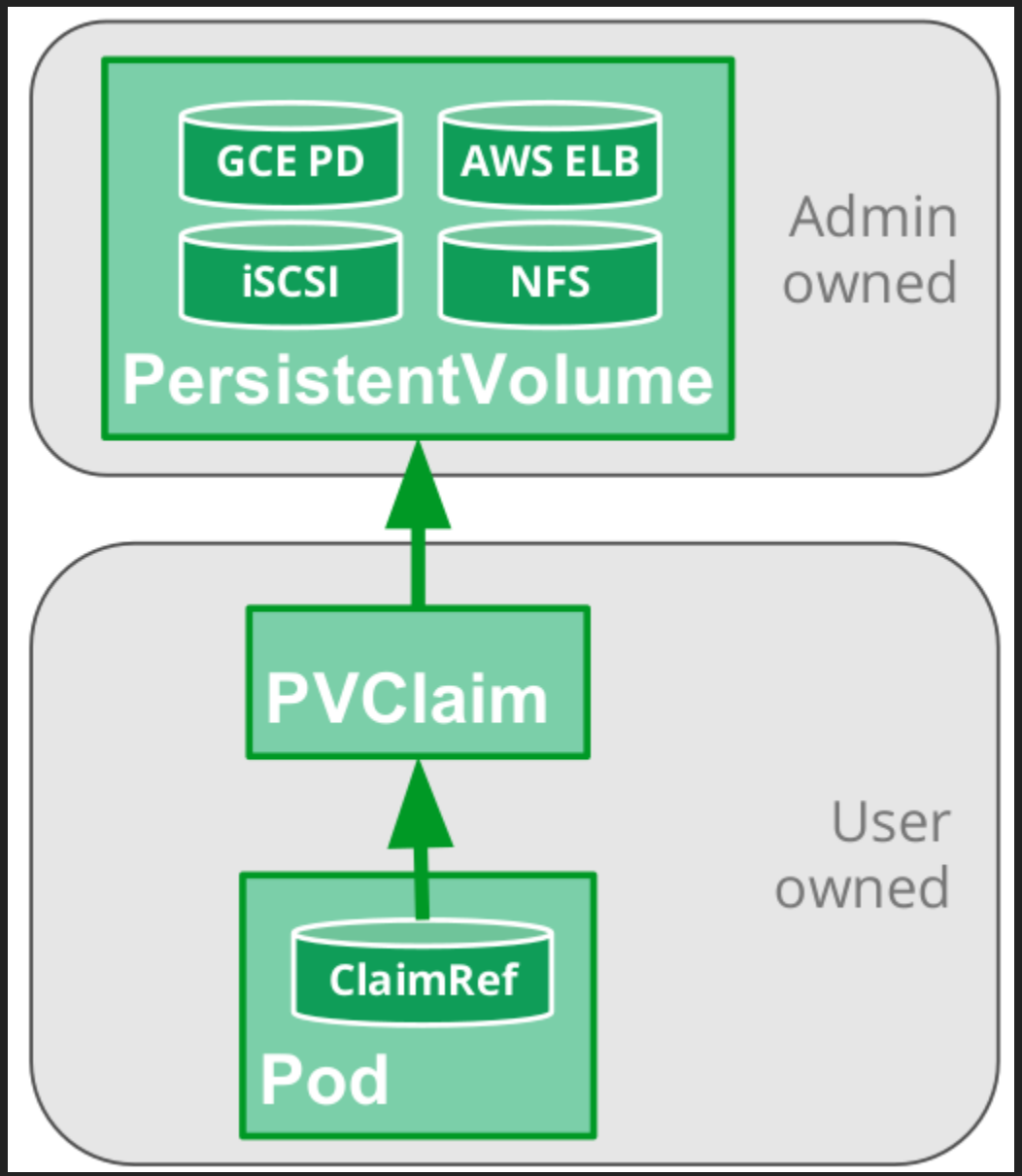
PODS


Node
Pod
Container
Container
Container
Container
Pod
Container
Container
Container
Container

SERVICES


Namespace
Service
RC
Container
Container
Pod
Container
Container
Pod
Service
RC
Container
Pod
Container
Pod

PERSISTENT VOLUMES


GCE PD
AWS ELB
iSCSI
NFS
PersistentVolume
Admin owned
PVClaim
ClaimRef
Pod
User owned

GCEPersistentDisk	CephFS
AWSElasticBlockStore	Cinder (OpenStack block storage)
AzureFile	Glusterfs
AzureDisk	VsphereVolume
FC (Fibre Channel)	Quobyte Volumes
FlexVolume	HostPath
Flocker	VMware Photon

NFS	vPortworx Volumes
iSCSI	ScaleIO Volumes
RBD (Ceph Block Device)	StorageOS

ACCESS MODES

- **ReadWriteOnce** – the volume can be mounted as read-write by a single node
- **ReadOnlyMany** – the volume can be mounted read-only by many nodes
- **ReadWriteMany** – the volume can be mounted as read-write by many nodes

USAGE

- Provision Disk
- Create Persistent Volume
- Create Persistent Volume Claim
- Mount Claim in Pod

PERSISTENT VOLUME

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0003
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Recycle
  nfs:
    path: /tmp
    server: 172.17.0.2
```

PERSISTENT VOLUME CLAIM

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
```

MOUNTING A PVC

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
    - name: mydatabase
      image: mysql:5.7
      volumeMounts:
      - mountPath: "/var/lib/mysql"
        name: mypd
  volumes:
    - name: mypd
      persistentVolumeClaim:
        claimName: myclaim
```

COMMON PITFALLS

- The abstraction is leaky.
- Local storage vs. SAN storage vs. Network Filesystems

AUTO-PROVISIONING

AUTO-PROVISIONING

- Manual creation of Persistent Volumes is error-prone.
- Does not scale well.

STORAGECLASS

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: standard
provisioner: kubernetes.io/aws-ebs
parameters:
  type: gp2
```

USING A STORAGECLASS

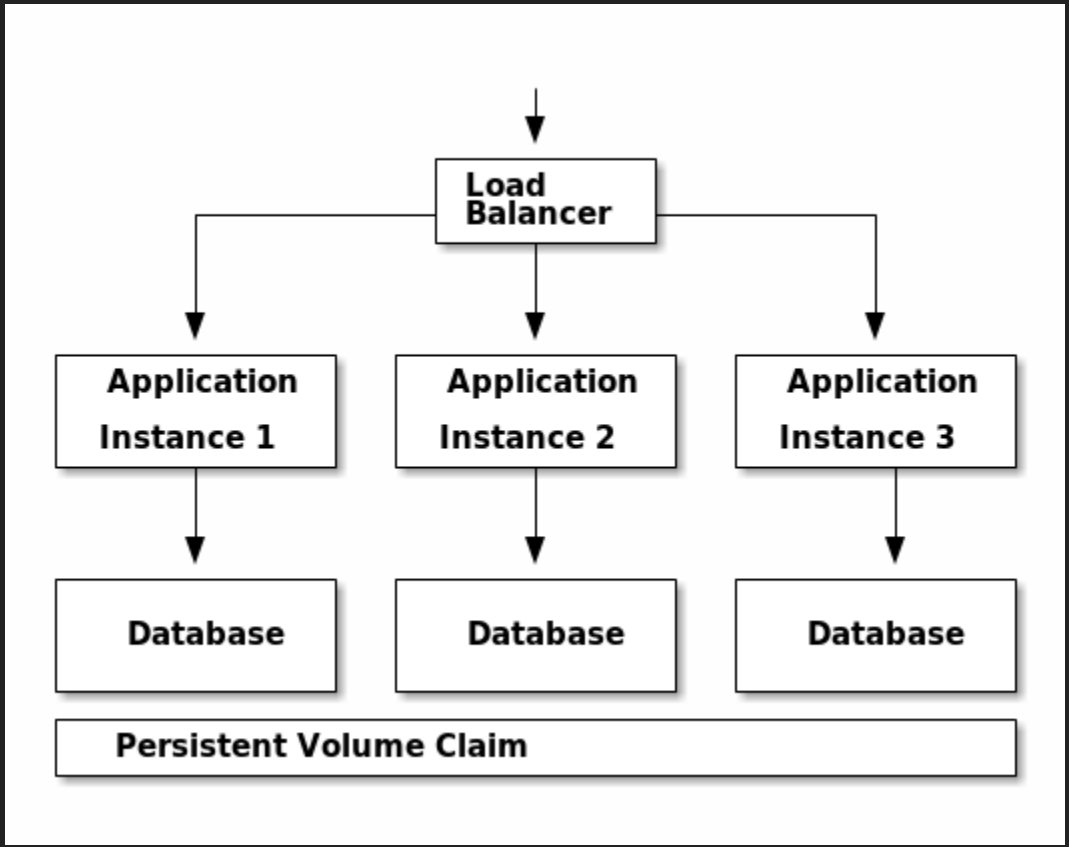
```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: myclaim
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
  storageClassName: standard
```

SUPPORTED PROVISIONERS

- default volume plugins (in-tree provisioners)
- out-of-tree provisioners
  - community provisioners: https://github.com/kubernetes-incubator/external-storage
  - write your own: https://github.com/kubernetes-incubator/external-storage/tree/master/docs/demo/hostpath-provisioner

SCALING

ARCHITECTURE REVISITED



SHARING PVCS

- Need different paths per Pod.
- Share needs to be available on every host.
- Corruption of file system affects all database nodes.
- Sharing a Persistent Volume Claim is not a good idea.

STATEFUL SETS

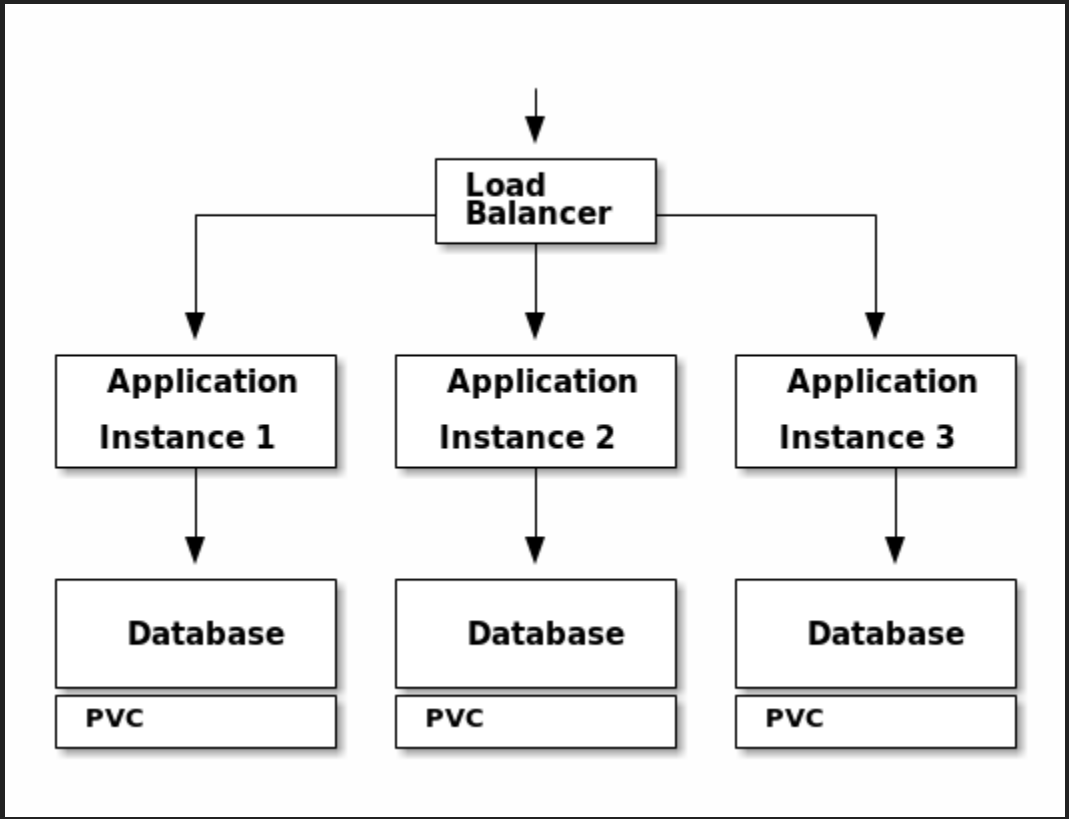
STATEFULSETS

- Pod template mechanism
- Hostnames are atomically increased:
  - pod-0
  - pod-1
  - ...
- Volume Claims can be provisioned on-the-fly

STATEFULSET EXAMPLE

```
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: gcr.io/google_containers/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
```

ARCHITECTURE STATEFULSETS



WRAP UP

LINKS

- Persistent Volumes: https://kubernetes.io/docs/concepts/storage/persistent-volumes
- Tutorial StatefulSet: https://kubernetes.io/docs/tutorials/stateful-application/basic-stateful-set/
- Demos: https://goo.gl/NCnzq8

THE END

@NicolasByl