STORAGEOS

How to migrate databases into Kubernetes

Alex Chircop & Ferran Castell

About US

Alex Chircop

- Founder & CEO of StorageOS/Co-chair CNCF Storage SIG
- Building a software defined, cloud native storage platform
- 25 years of engineering infrastructure platforms before embarking on the startup adventure
- Previously Goldman Sachs and Nomura
- Twitter: @chira001

Ferran Arau Castell

- Product Reliability Engineer at StorageOS
- Building and running 24/7 platforms for over 10 years
- Twitter: @FerranArau

STORAGEOS

StorageOS is cloud native, software-defined storage for running containerized applications in production, running in the cloud, on-prem and in hybrid/multi-cloud environments.

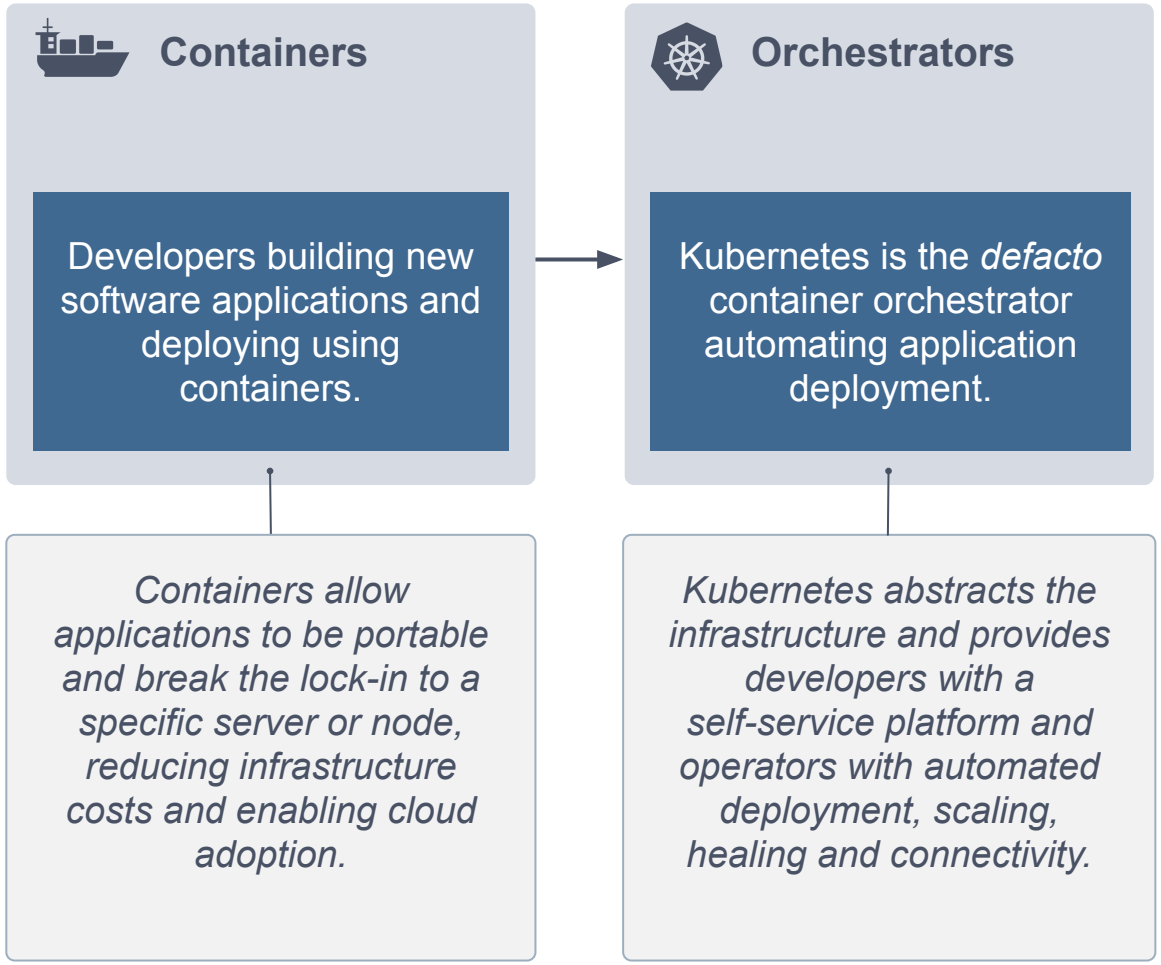
The Journey to Cloud Native

Containers

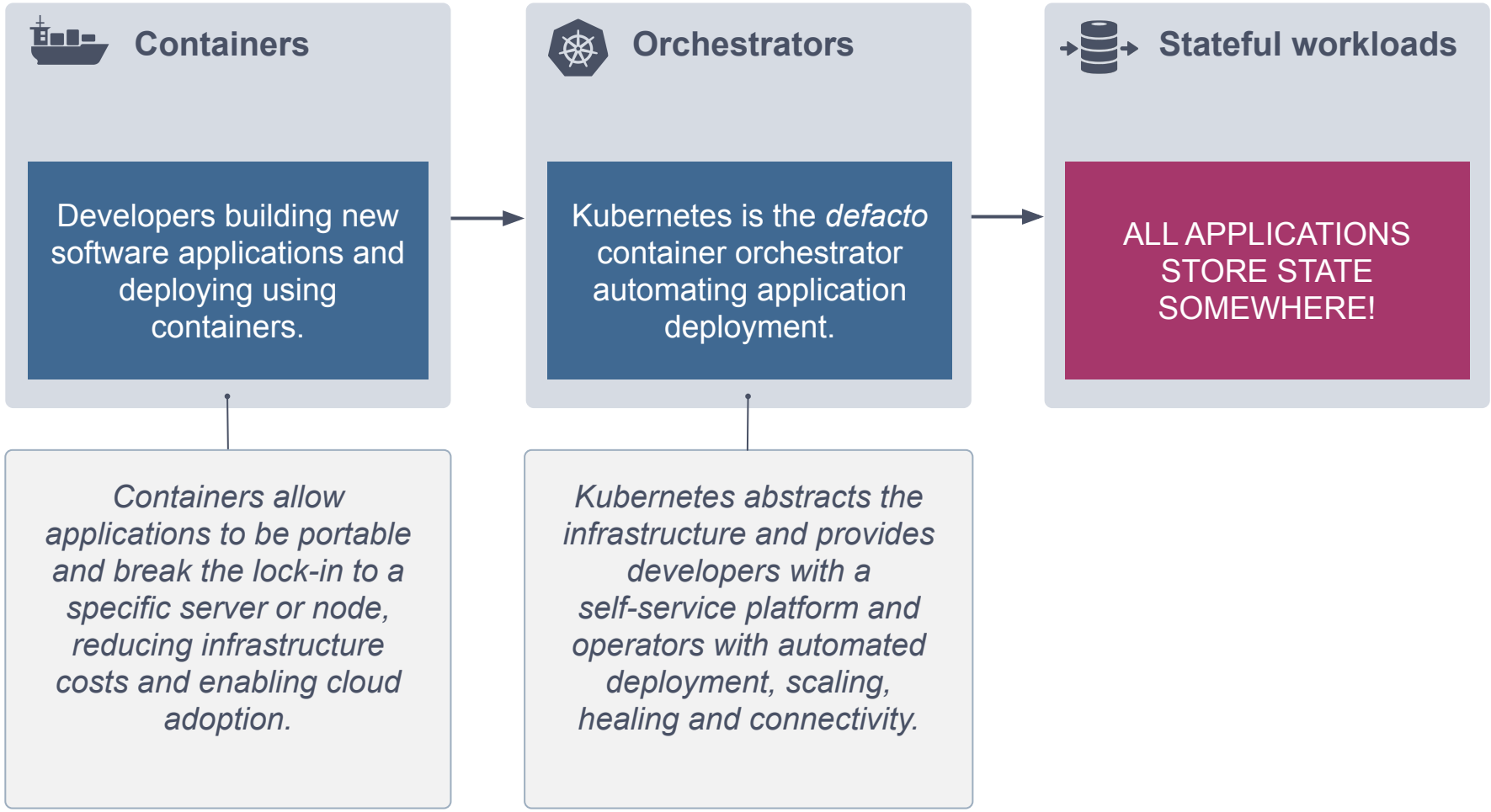
Developers building new software applications and deploying using containers.

Containers allow applications to be portable and break the lock-in to a specific server or node, reducing infrastructure costs and enabling cloud adoption.

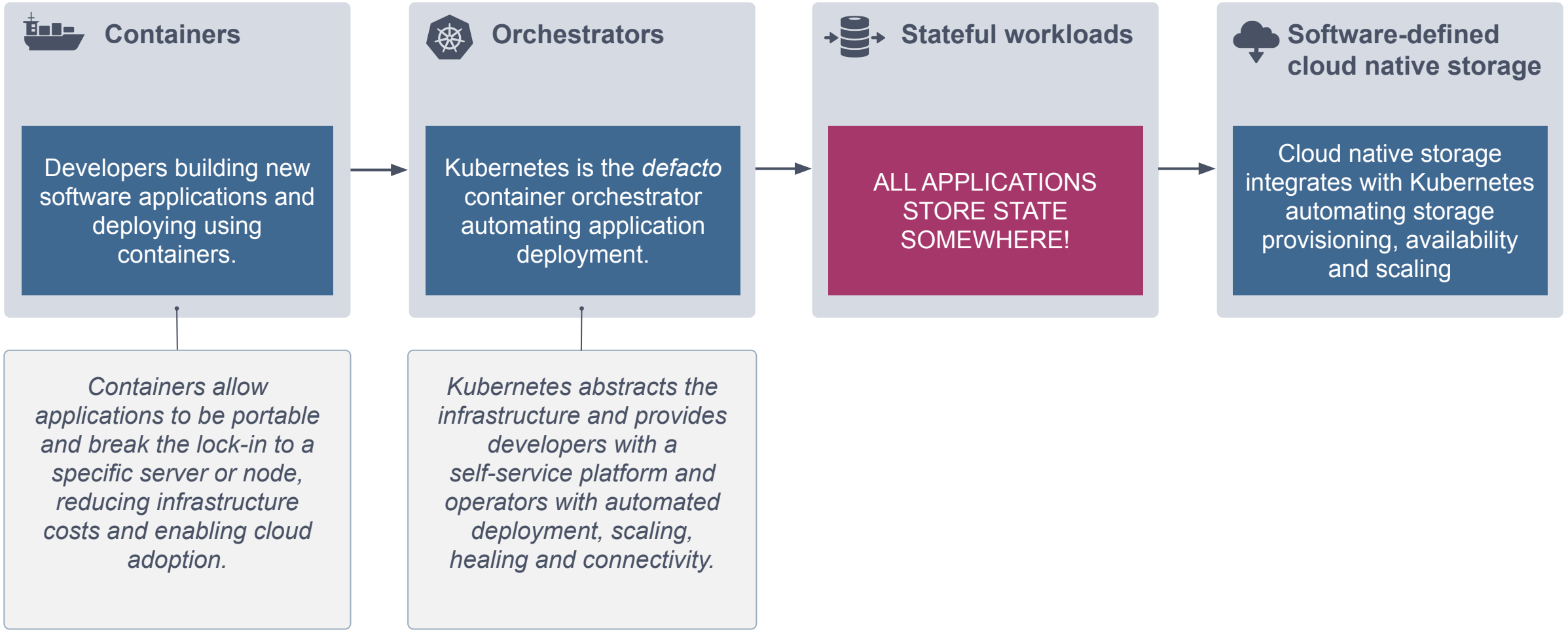
The Journey to Cloud Native



The Journey to Cloud Native



The Journey to Cloud Native



The Journey to Cloud Native

Containers

Developers building new software applications and deploying using containers.

Containers allow applications to be portable and break the lock-in to a specific server or node, reducing infrastructure costs and enabling cloud adoption.

Orchestrators

Kubernetes is the *defacto* container orchestrator automating application deployment.

Kubernetes abstracts the infrastructure and provides developers with a self-service platform and operators with automated deployment, scaling, healing and connectivity.

Stateful workloads

ALL APPLICATIONS STORE STATE SOMEWHERE!

Software-defined cloud native storage

Cloud native storage integrates with Kubernetes automating storage provisioning, availability and scaling

Application Centric Storage

Move all applications to apply the benefits of orchestration:

Build stateful applications

Enable self service storage provisioning:

Build anything as a service

Dynamic Provisioning in K8s: How it Works

Show me the YAML!

Dynamic Provisioning in K8s: How it Works

Show me the YAML!

StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: storageos
 labels:
   app: storageos
provisioner: csi.storageos.com
parameters:
 fsType: ext4
 pool: default
 adminSecretNamespace: default
 adminSecretName: storageos-api
```

Dynamic Provisioning in K8s: How it Works

Show me the YAML!

StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: storageos
 labels:
   app: storageos
provisioner: csi.storageos.com
parameters:
 fsType: ext4
 pool: default
 adminSecretNamespace: default
 adminSecretName: storageos-api
```


PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: database-vol1
spec:
 storageClassName: storageos
 accessModes:
   - ReadWriteOnce
 resources:
   requests:
     storage: 5Gi
```

Dynamic Provisioning in K8s: How it Works

Show me the YAML!

StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: storageos
 labels:
   app: storageos
provisioner: csi.storageos.com
parameters:
 fsType: ext4
 pool: default
 adminSecretNamespace: default
 adminSecretName: storageos-api
```

PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: database-vol1
spec:
 storageClassName: storageos
 accessModes:
   - ReadWriteOnce
 resources:
   requests:
     storage: 5Gi
```

Application (Pod/StatefulSet)

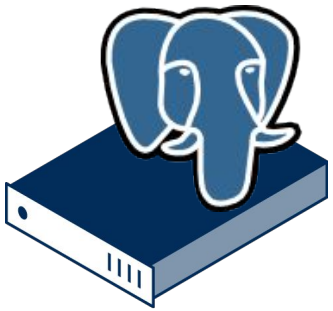
```
apiVersion: v1
kind: pod
metadata:
name: postgres
spec:
containers:
- name: pg
  image:
crunchydata/crunchy-postgres:centos7
-10.4-1.8.3
  volumeMounts:
    - mountPath: /pgdata
      name: data
volumes:
- name: data
  persistentVolumeClaim:
    claimName: database-vol1
```

Cloud Native Journey for a Database

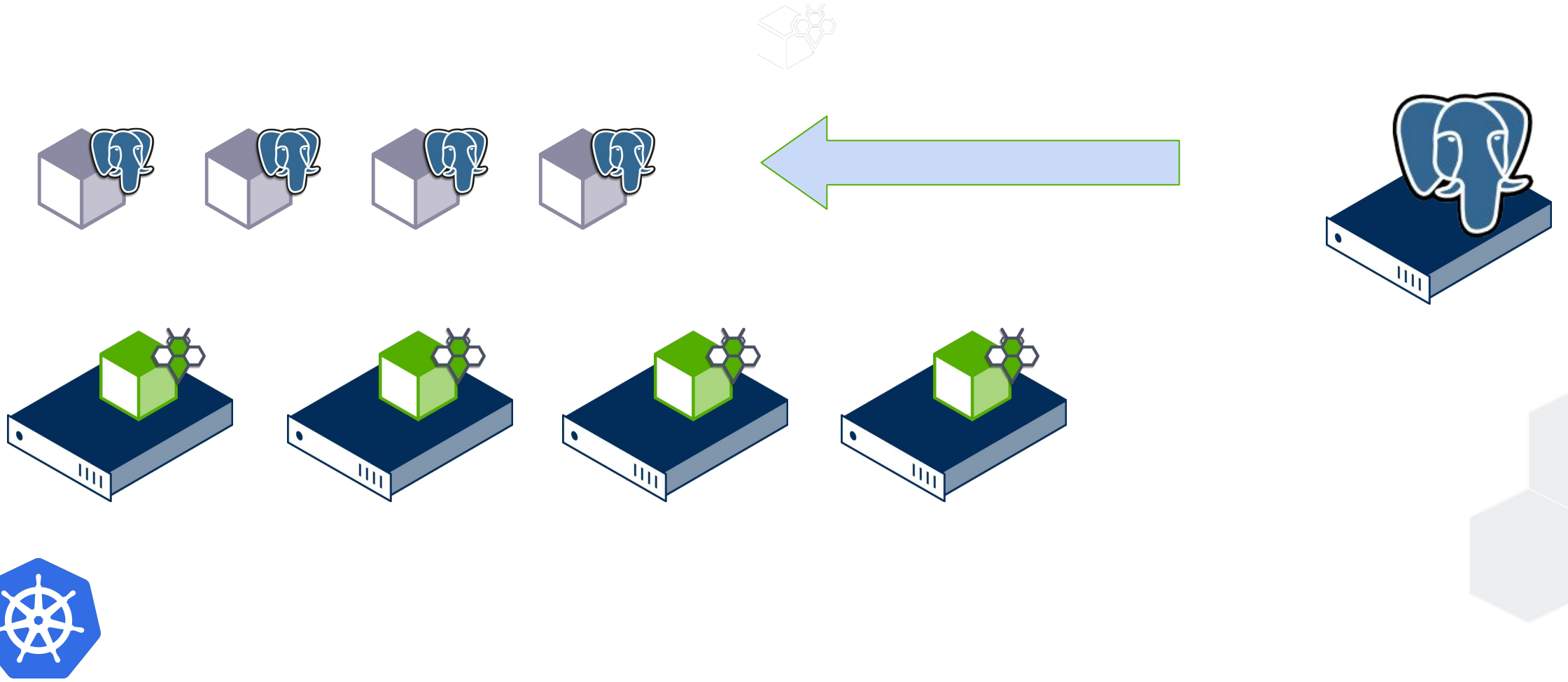
Server Centric

Scale-up Architecture

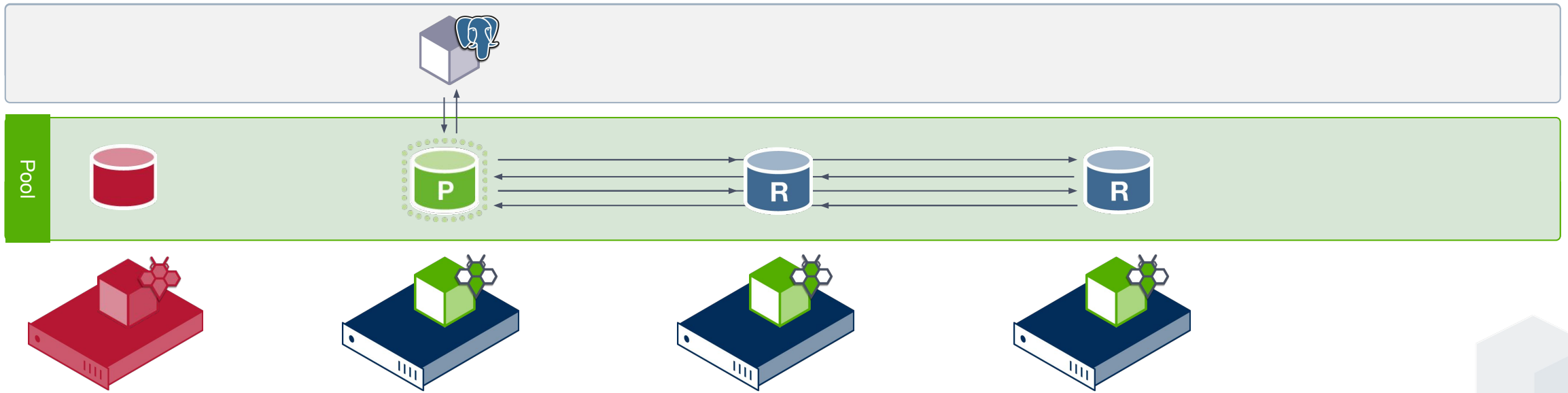
Multiple databases on a node



Cloud Native Journey for your Database



Replication & Automatic Failover



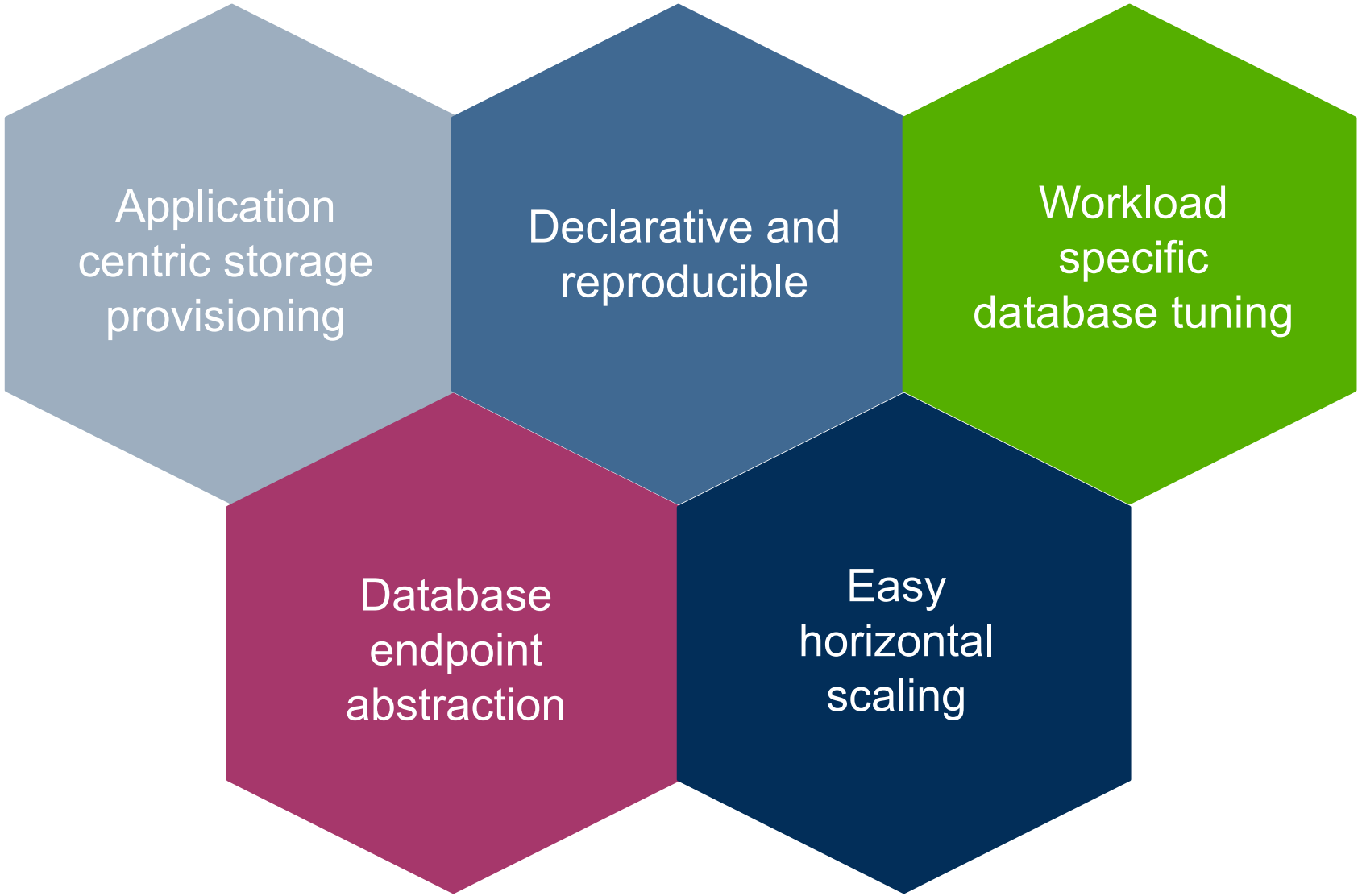
Stateful Applications in Kubernetes

- Stateful vs stateless
- Containers are ephemeral, nodes can be ephemeral
- Pods scheduled anywhere, rescheduled somewhere else
- Performance vs flexibility
- Capacity provisioning

- Multiples copies of data with automatic failover
- Access data from anywhere
- Low latency, high throughput and predictable performance
- Application centric

Live Demo

Database-as-a-Service



How Can I Try StorageOS?

StorageOS **Developer Edition** is forever FREE with 5TB storage

Docs site - https://docs.storageos.com/docs/support/

Self Evaluation Guide - https://docs.storageos.com/docs/self-eval/

Join us on Slack - https://slack.storageos.com/

Email us info@storageos.com

Questions

STORAGEOS

Thank You

www.storageos.com