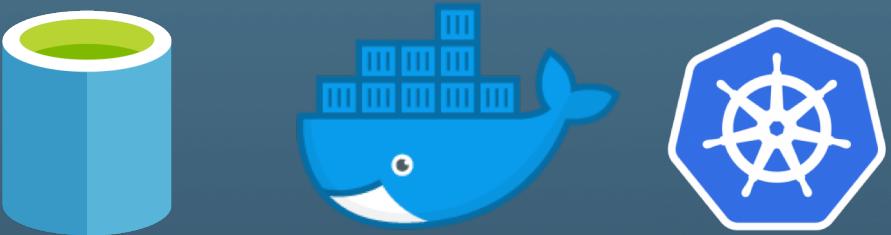


# DBA MASTERY



Getting started SQL Containers  
Docker and Kubernetes





# Carlos Robles

Principal Consultant, DBA Mastery



/croblesdba



@dbamastery



crobles@dbamastery.com

## Experience

Microsoft Data Platform MVP

Friend of Redgate

+10 years of experience multi-platform  
Solutions architect

## Community

GTSSUG – Community Leader

Speaker, author, blogger, mentor

Simple Talk, SQL Server Central & MSSQL Tips

## DBA Mastery

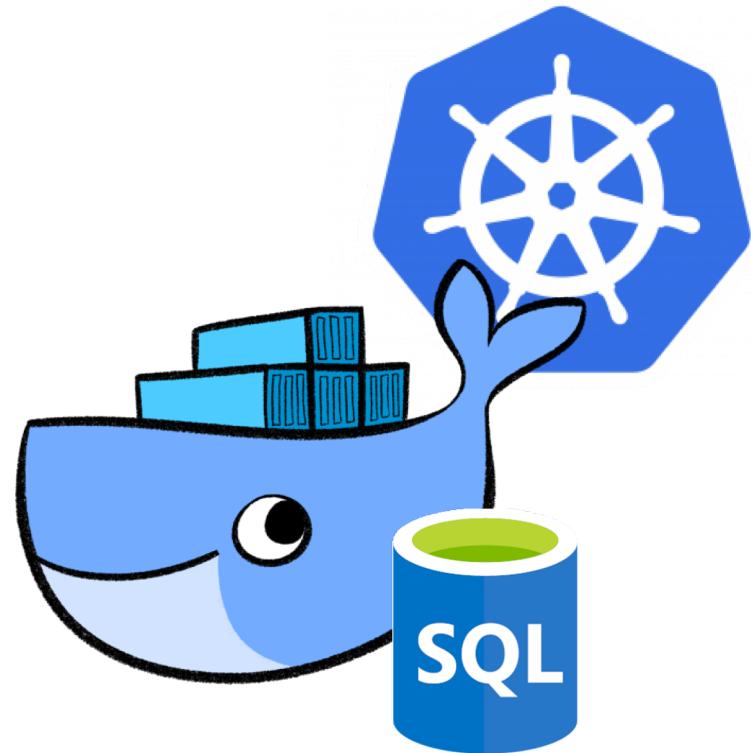
Tips, scripts, best practices & more



MAXDOP Calculator  
Azure Data Studio wait stats widget  
Azure Data Studio notebooks  
PerfMon for DBAs - PowerShell

# Agenda

- Introduction to Docker
- SQL Server containers
- Introduction to Kubernetes
- Demos ...



# Introduction to Docker



# What is Docker?



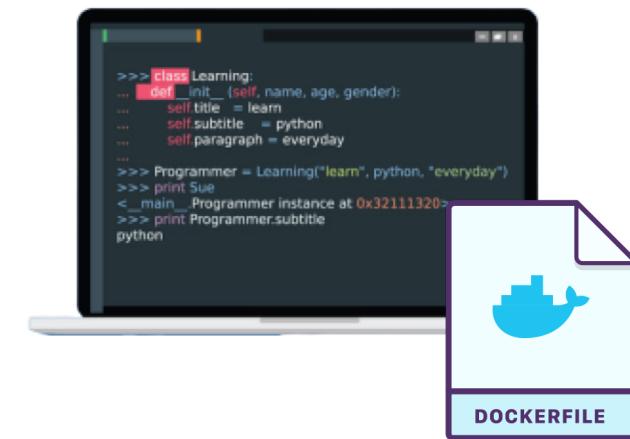
*Docker is an open platform for developing, shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly.*

*With Docker, you can manage your infrastructure in the same ways you manage your applications.*



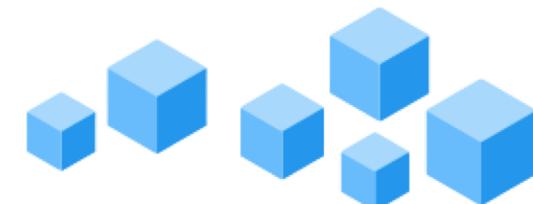
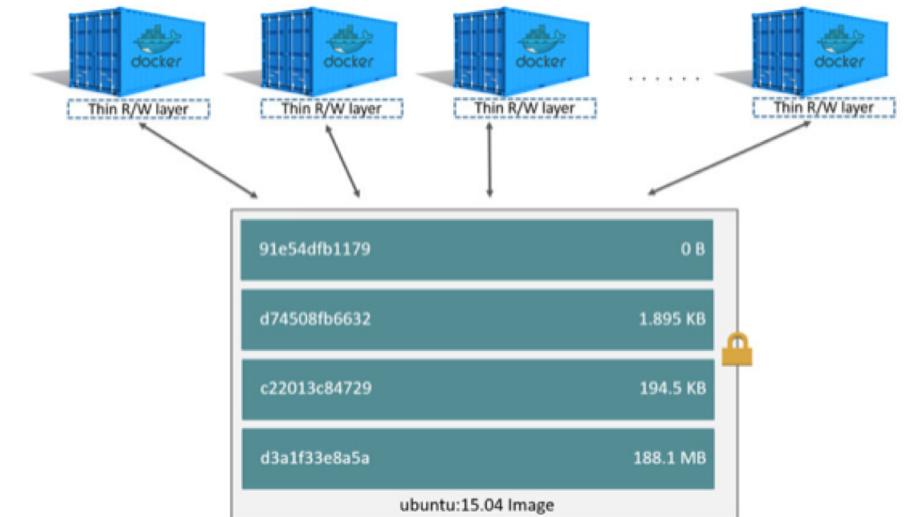
# Images

- Binary file with instructions
- Created with a Dockerfile
- Each instruction = layer
- Can be created from existing image

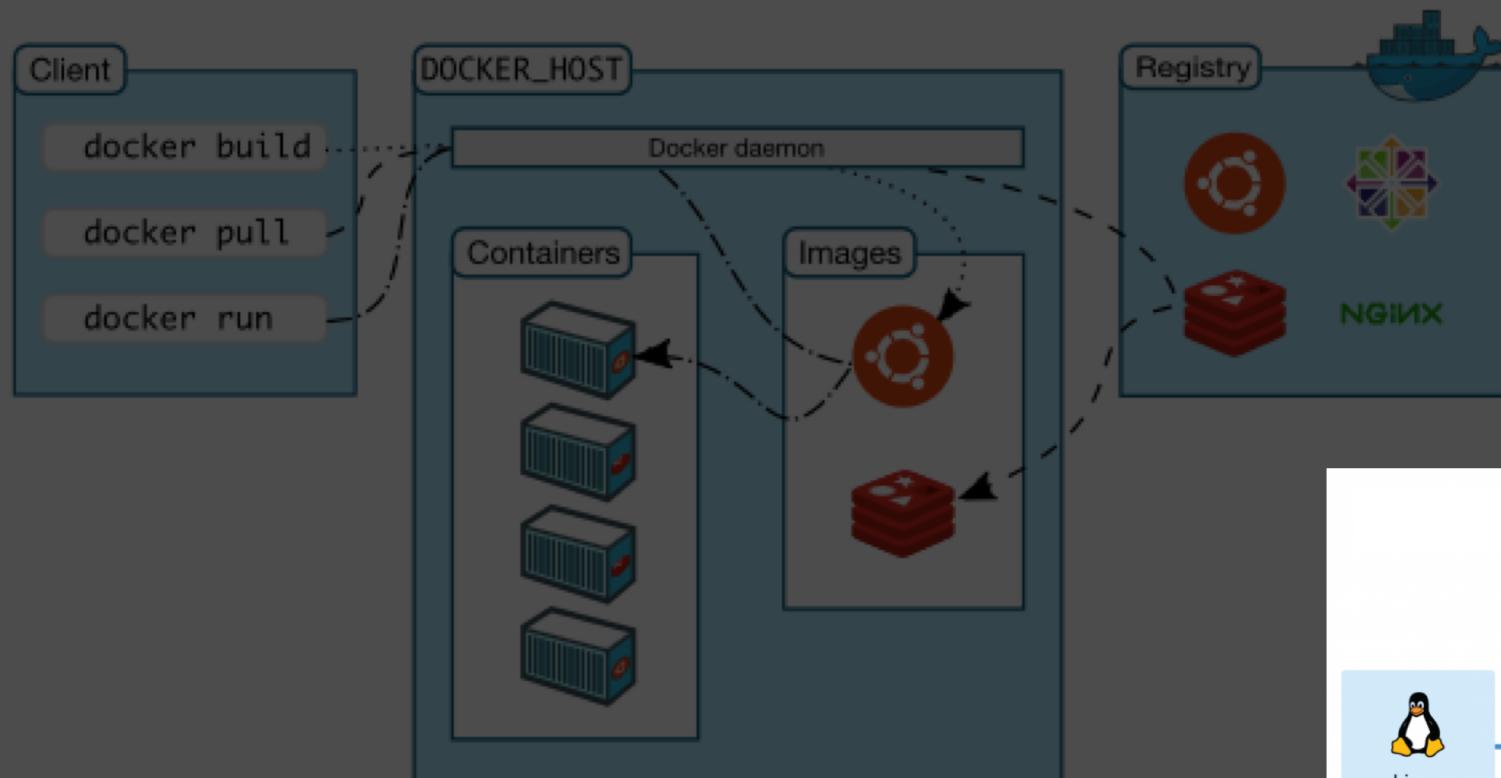


# Containers

- Runnable instance of an image
- Just a program / process
- Writable and shared read-only layer
- Small storage footprint
- Persistent storage = volumes



# Docker architecture



# VM's vs Containers

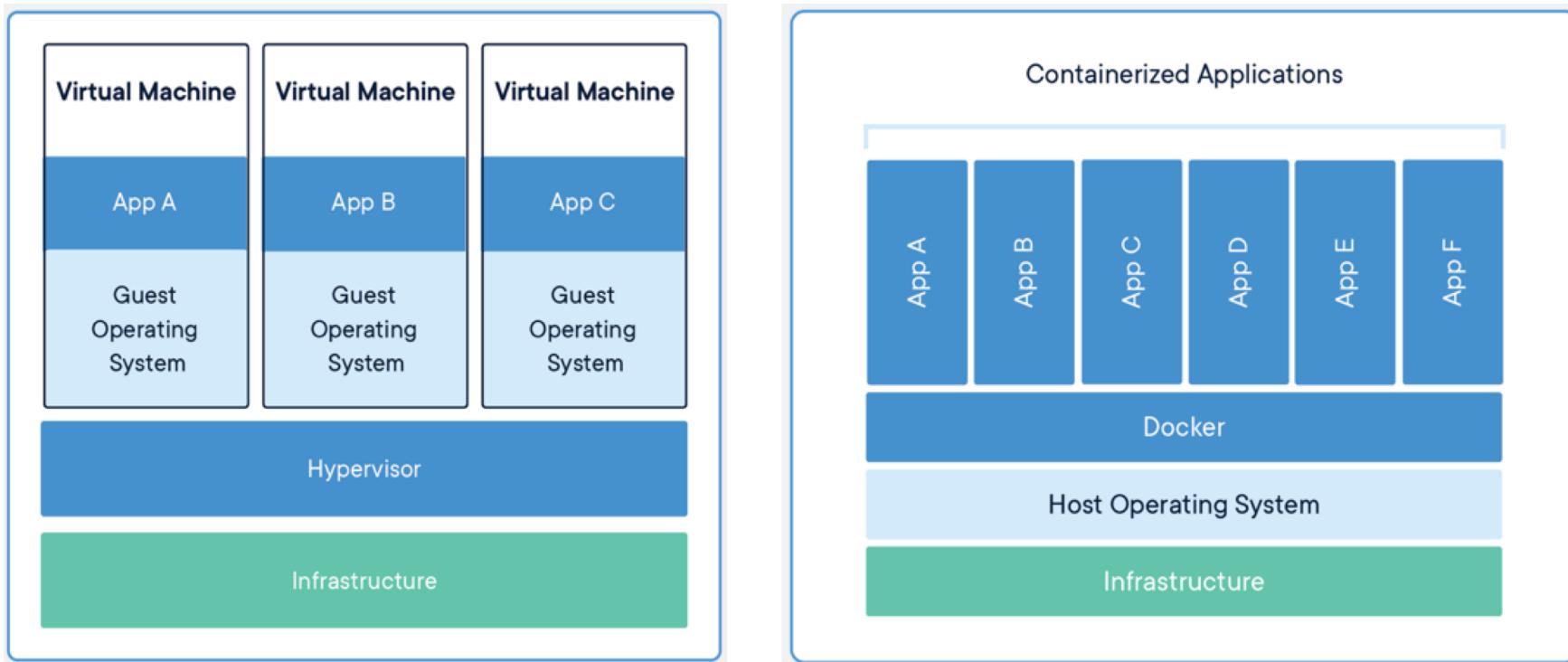


- 
- Virtualization +15 years
  - Heavyweight
  - Hardware virtualization
  - Each VM = entire OS



- 
- No installation
  - Lightweight
  - OS virtualization
  - All containers = same host OS



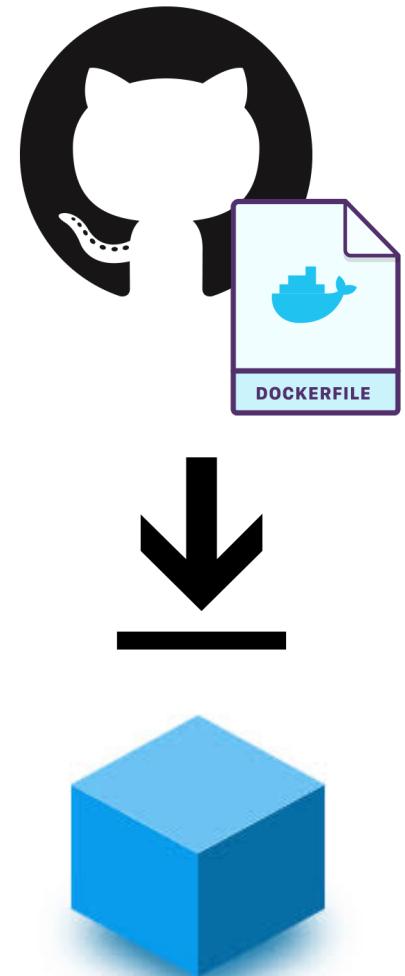


# SQL Server containers



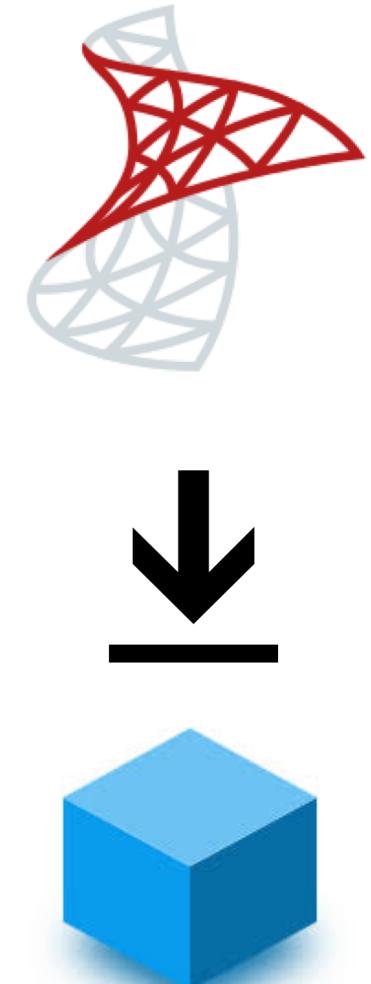
# SQL Server Dockerfile

```
FROM ubuntu:16.04  
  
EXPOSE 1433  
  
COPY ./install /  
  
CMD ["/opt/mssql/bin/sqlservr"]
```



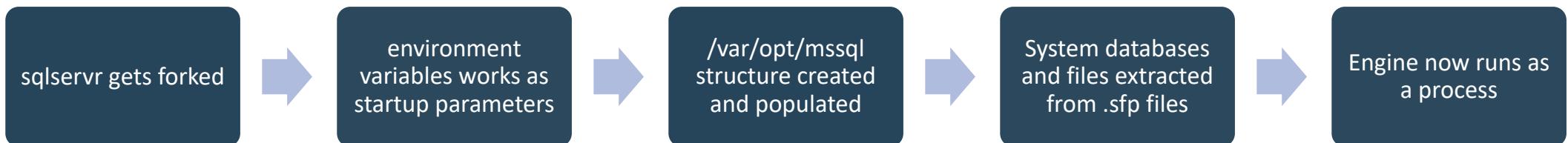
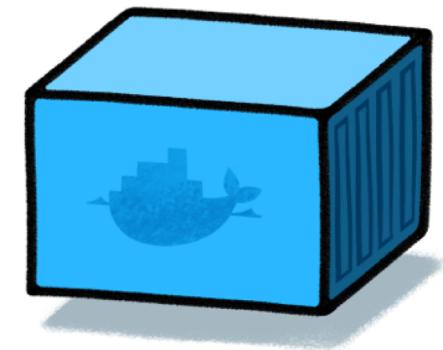
# SQL Server image

- [Docker Hub – Microsoft container registry](#)
- SQL Server 2017
  - Just Ubuntu from RTM to latest CU
- SQL Server 2019
  - Ubuntu and RedHat
  - From CTP to latest CU



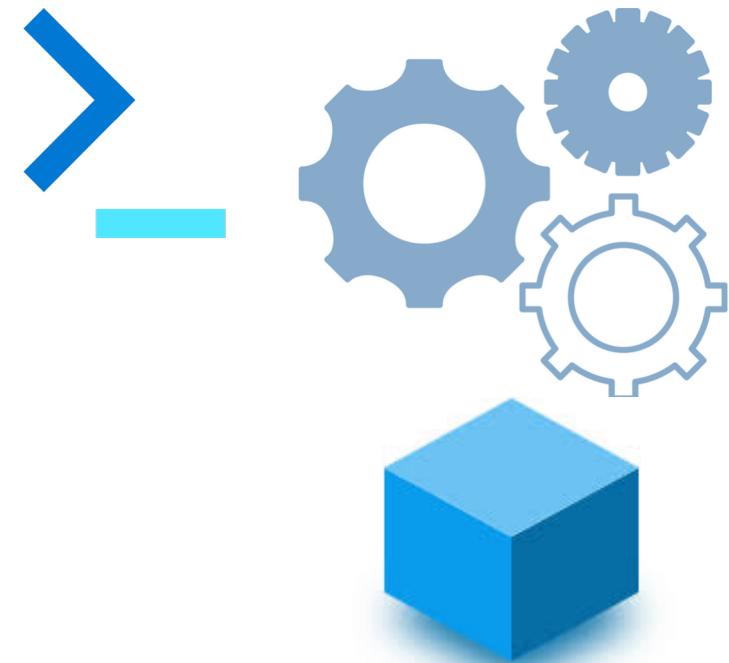
# My first SQL Container

```
docker run \
--name SQL-01 \
--env 'ACCEPT_EULA=Y' \
--env 'MSSQL_SA_PASSWORD=_SqLr0ck5_' \
--publish 1433:1433 \
--detach mcr.microsoft.com/mssql/server:2019-CU6-ubuntu-18.04
```



# Docker CLI

```
docker pull  
docker run  
docker start | stop  
docker image | container  
docker rm | rmi  
docker exec  
docker build  
docker logs  
docker inspect  
docker volume  
docker save
```



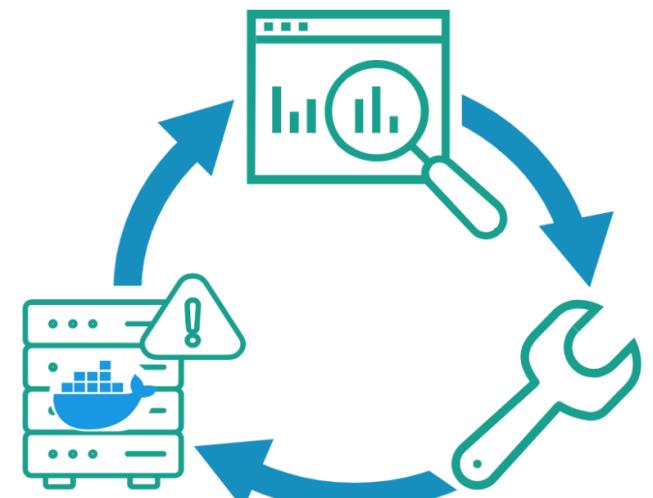
# Demo

Docker + SQL Containers  
(The basics)



# Use cases

- Local development
- Troubleshooting
- Demonstrations
- Eliminates shared environments
- Eliminates resource contention
- DevOps



# Introduction to Kubernetes



# What is Kubernetes?



kubernetes

*Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.*

*It also make possible the container orchestration for automating application deployment, scaling, and management.*



# The basics

- Master
  - Multiple moving parts \ processes
  - Runs on a single node in the cluster
  - Tells what to do – desired state
- Nodes
  - Do the work, runs applications
  - Aka “minions”
  - Reports the state back up to the master



- Pods
  - Containers runs inside of pods
  - Can have one or more pods within a node
- Services
  - Hiding multiple pods behind a service IP address
- Deployments
  - Declarative model
    - Desired state (number of POD's)
  - Manifest file (YAML, JSON)



# SQL Server deployment

- Create a namespace
- Create a secret for the sa password
- Create a PVC
- Create a Load balancer (Service)
- Create a SQL Server deployment
- Check deployment
- Get events, pod name
- Describe the pod
- Get the logs
- Get IP address
- Connect to SQL Server
- Enjoy 😊



## Service

```
apiVersion: v1
kind: Service
metadata:
  name: mssql-service
spec:
  selector:
    app: mssql
  ports:
    - protocol: TCP
      port: 31433
      targetPort: 1433
  type: LoadBalancer
```

associate with pod

connect with this port

## Persistent volume claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mssql-data
annotations:
  volume.beta.kubernetes.io/storage-
class: managed-premium
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 8Gi
```

Local or cloud

Read/Write one node at a time

Capacity

# Deployment

```
apiVersion: apps/v1
```

```
kind: Deployment
```

```
metadata:
```

```
  name: mssql-deployment
```

```
spec:
```

```
  replicas: 1
```

keep 1 always running

```
  selector:
```

```
    matchLabels:
```

```
      app: mssql
```

associate with  
LoadBalancer

```
  strategy:
```

```
    type: Recreate
```

```
template:
```

```
  metadata:
```

```
    labels:
```

```
      app: mssql
```

recreate pod  
when updated

convenient way

to query and manage pod

```
spec:
```

```
  terminationGracePeriodSeconds: 10
```

```
  securityContext:
```

```
    fsGroup: 1000
```

give non-root user permissions to PVC

```
  containers:
```

```
- name: mssql
```

```
  image: mcr.microsoft.com/mssql/rhel/server:2019-latest
```

```
  env:
```

```
- name: MSSQL_PID
```

```
  value: "Developer"
```

```
- name: ACCEPT_EULA
```

```
  value: "Y"
```

```
- name: MSSQL_SA_PASSWORD
```

```
  valueFrom:
```

```
    secretKeyRef:
```

```
      name: mssql-secret
```

```
      key: SA_PASSWORD
```

add other environment  
variables here

your password

```
  volumeMounts:
```

```
- name: mssqldb
```

```
  mountPath: /var/opt/mssql
```

SQL databases

```
  volumes:
```

```
- name: mssqldb
```

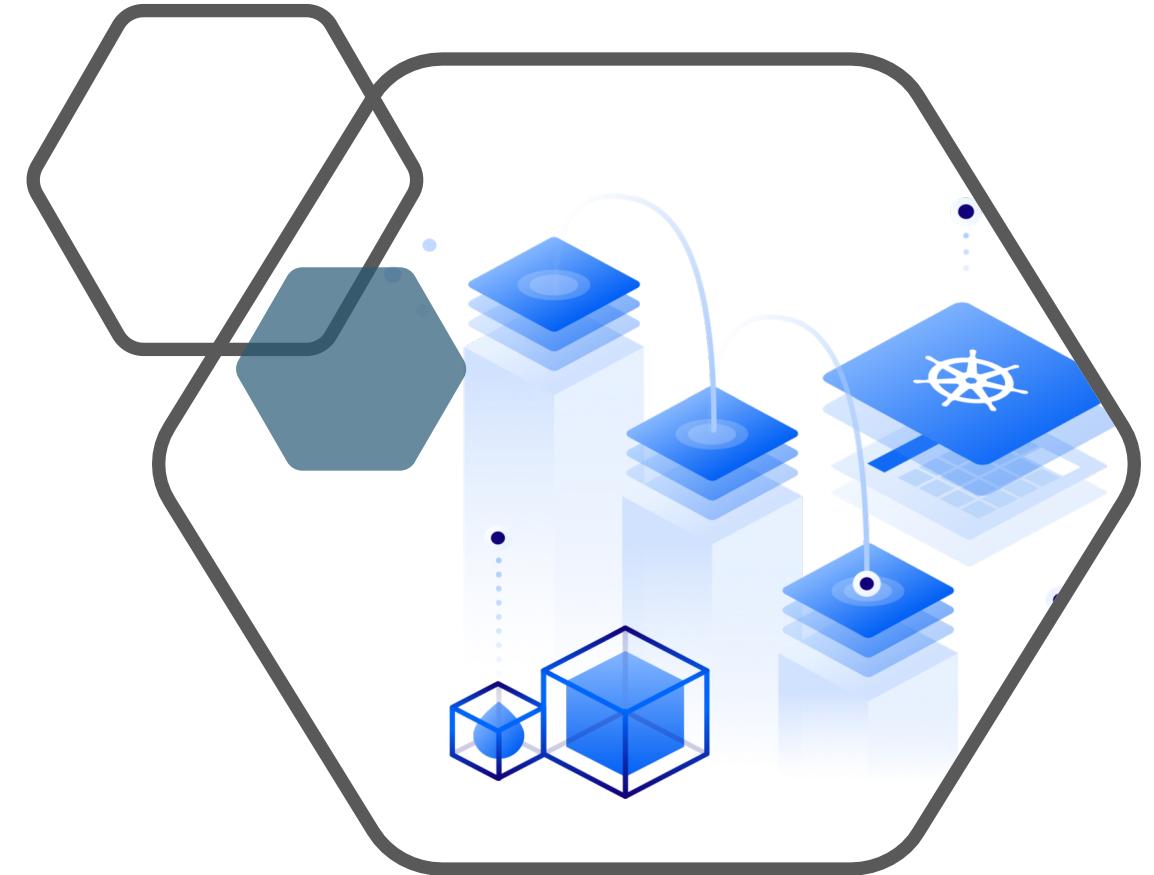
```
  persistentVolumeClaim:
```

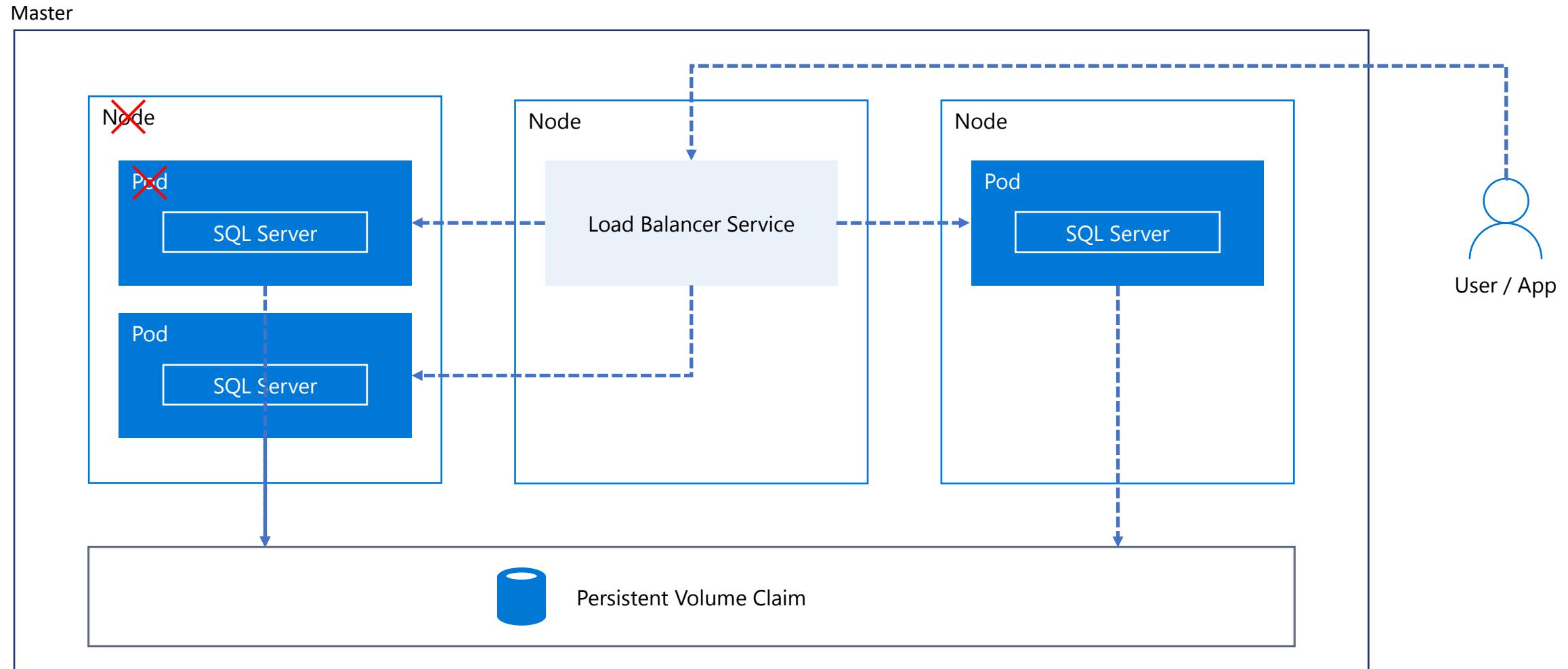
```
    claimName: mssql-data
```

PVC

# Demo

Kubernetes + SQL Containers  
(The basics)





# Questions?



@dbamastery



@dbamaster



# References

- **Official documentations**
  - [Docker Docs](#)
  - [Kubernetes Docs](#)
- **SQL Server**
  - [SQL Server workshops](#)
  - [SQL Server in Docker](#)
- **Books**
  - [The Phoenix / Unicorn Project](#)
  - [The DevOps Handbook](#)
  - [Kubernetes: Up and Running](#)
  - [Designing Distributed Systems](#)
  - [Pro SQL Server On Linux by Bob Ward](#)
- **Pluralsight**
  - [Getting Started with Docker](#)
  - [Docker Deep Dive](#)
  - [Docker and Kubernetes: The big picture](#)
  - [Kubernetes Installation and Configuration fundamentals](#)
- **Microsoft Learning**
  - [Kubernetes Learning Path](#)
  - [SQL Workshops](#)
- **Katacoda**
  - [Docker](#)
  - [Kubernetes](#)



# More about SQL Containers?

- Published articles
  - [Redgate - Simple Talk](#)
  - [SQL Server Central](#)
  - [MSQL Tips](#)



redgate





@dbamastery



@dbamaster



crobles@dbamastery.com



croblesdba

# Thanks!

