Exploring SQL Server containers on Docker and Kubernetes

Carlos Robles, Principal Consultant, DBA Mastery

Presenting Sponsors:

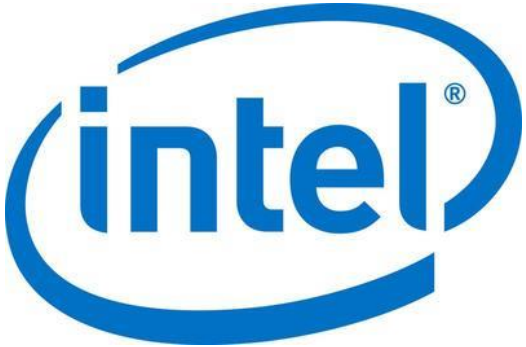
Technical Assistance

If you require assistance during the session, type your inquiry into the question pane on the right side.

Maximize your screen with the zoom button on the top of the presentation window.

Please fill in the short evaluation following the session. It will appear in your web browser.

Thank you to our Presenting Sponsors



Explore everything PASS has to offer

Free Online Resources

Newsletters

PASS.org

PASS SUMMIT

PASS' flagship event
November 5-8
Seattle, Washington

PASS LOCAL GROUPS

Local user groups
around the world

PASS SQLSATURDAY

Free 1-day local
training events

PASS VIRTUAL GROUPS

Online special
interest user groups

PASS MARATHON

Business analytics
training

PASS VOLUNTEERS

Get involved

Carlos Robles

Principal Consultant, DBA Mastery

- /croblesdba
- @dbamastery
- Carlos Robles
- crobles@dbamastery.com

Experience

Over 10 years working with multiple DMBS

Microsoft Data Platform MVP

MCSE Data Management and Analytics

Community

Guatemala SQL Server User Group leader

International speaker, mentor, volunteer

MSSQL Tips and SQL Server Central author

DBA Mastery

SQL Server tips, best practices, scripts and more

MAXDOP calculator

Exploring SQL Server containers on Docker and Kubernetes

Carlos Robles, Principal Consultant, DBA Mastery

Presenting Sponsors:

Agenda

In this session we will explore the options we have when working with SQL Server running on Docker containers or Kubernetes.

- Introduction to Docker
- The SQL Server docker image
- The SQL Server Dockerfile
- How to start a SQL Server container
- Demo
- Introduction to Kubernetes
- Demo

Introduction to Docker

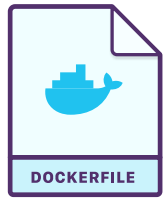
- From Docker docs:

  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.

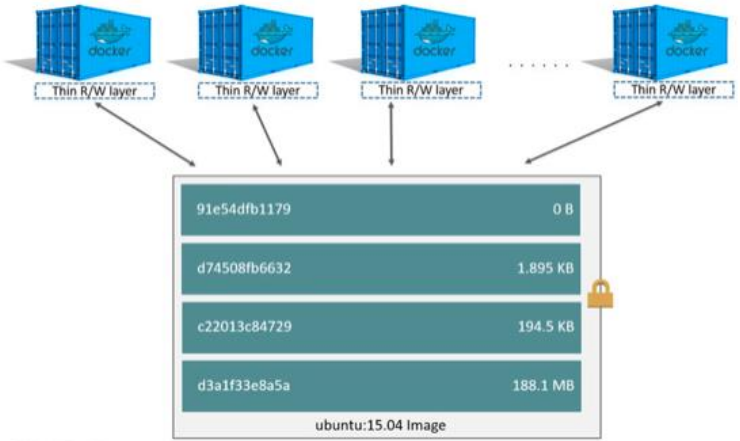
Images

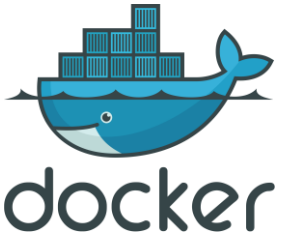
- Is a read-only template with instructions for creating a Docker container
- Images are created using a Dockerfile
- A snapshot of a set of files required to run an application
- A new image can be created from an existing image (make your own)
  - SQL Server for example, based on Ubuntu or RedHat
- Portable and consistent



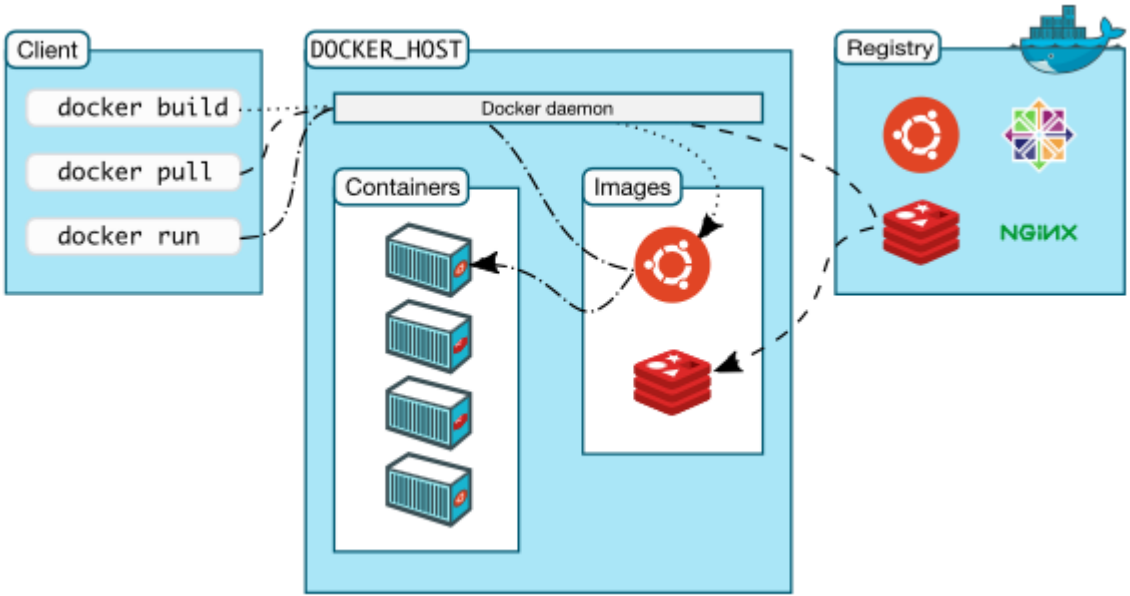
Containers

- The runnable instance of a Docker image
- A standardized unit of software
- Container is nothing more than a program
- Writable layer and shared read-only layer
  - Small storage footprint
- Containers has full access to all resources
- Volumes = Persistent storage





Docker architecture



VM’s vs Containers

Virtualization +15 years

Sometimes heavyweight

Hardware virtualization

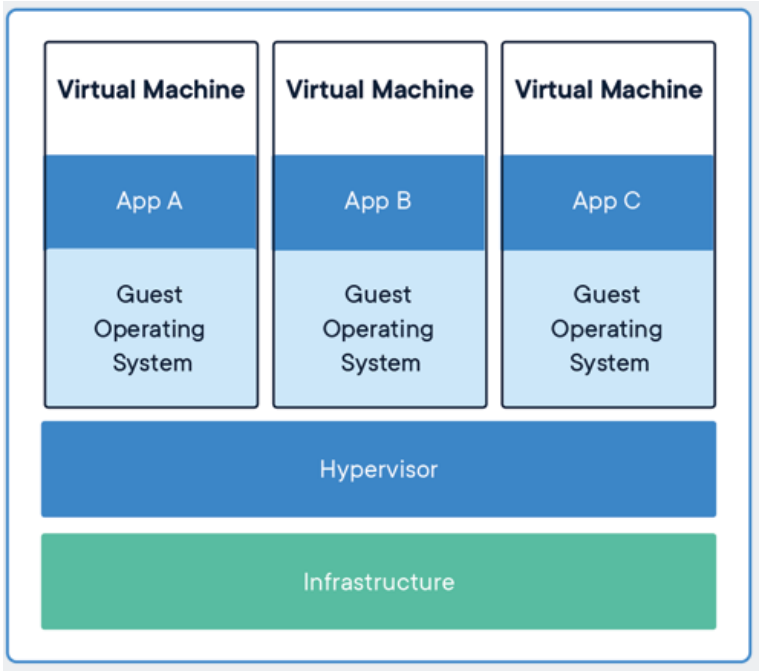
Each VM has an entire OS

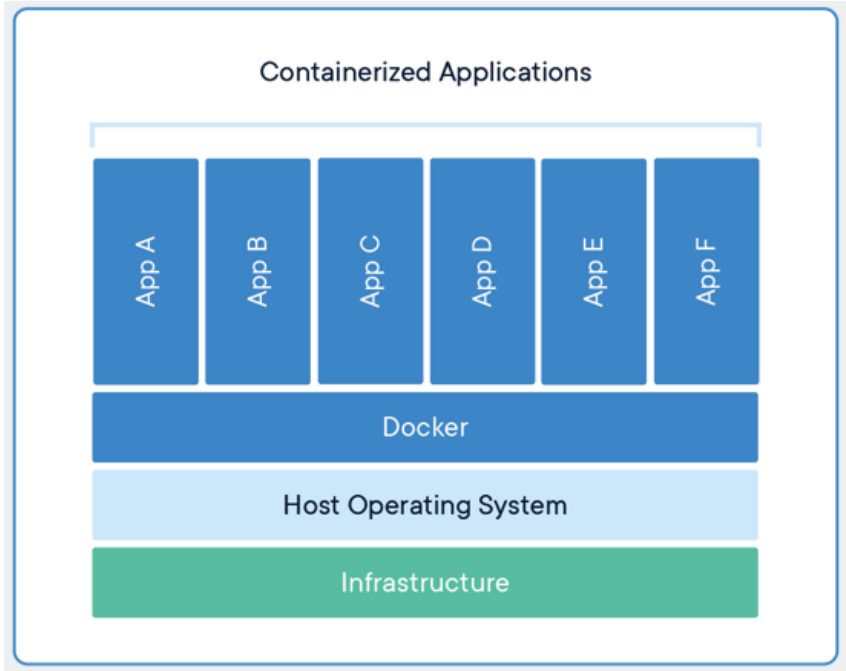
No installation

Lightweight

OS virtualization

All containers run in the same host OS

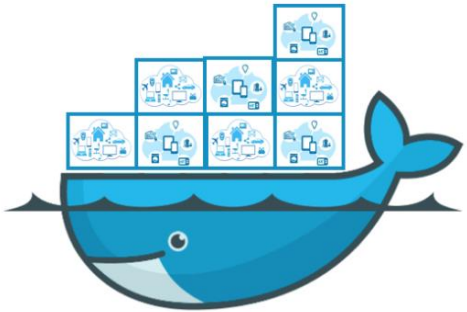




24HOURS OF PASS

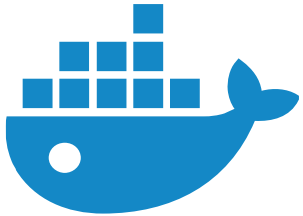
Advantages

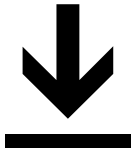
- Easy to use
- Agile application creation and deployment
- CI\CD - DevOps
- Environmental consistency across all platforms
- Cloud and OS portability
- Resource isolation
- Quick start \ stop time



The SQL Server docker image

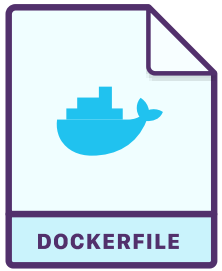
- Docker Hub – Microsoft container registry
- SQL Server 2017
  - Just Ubuntu from RTU to latest CU
- SQL Server 2019
  - Ubuntu and RedHat
  - From RTM to latest CU
- SQL Server is pre-installed (standard)
- Backups are compatible between all platforms

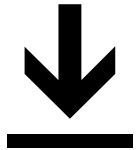




The SQL Server Dockerfile

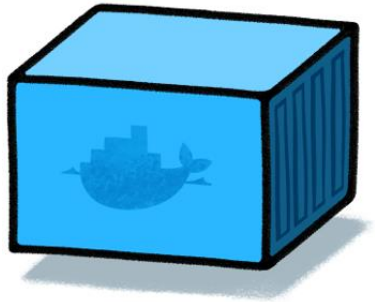
```
FROM ubutu:16.04

EXPOSE 1433

COPY ./install /

CMD ["/opt/mssql/bin/sqlservr"]
```





How to start a SQL Server container

```
docker run \
--name 24hop \
--env 'ACCEPT_EULA=Y' \
--env 'MSSQL_SA_PASSWORD=24hop#’ \
--publish 1400:1433\
--detach mcr.microsoft.com/mssql/server:2017-CU12
```



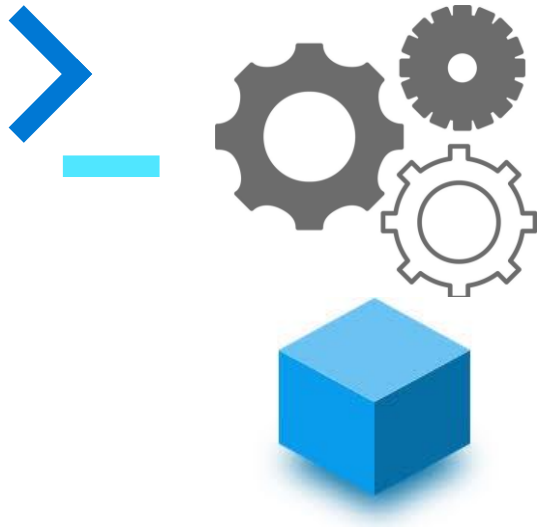
sqlservr gets forked → environment variables works as startup parameters → /var/opt/mssql structure created and populated → System databases and files extracted from .sfp files → Engine now runs as a process

Demo

Microsoft intel

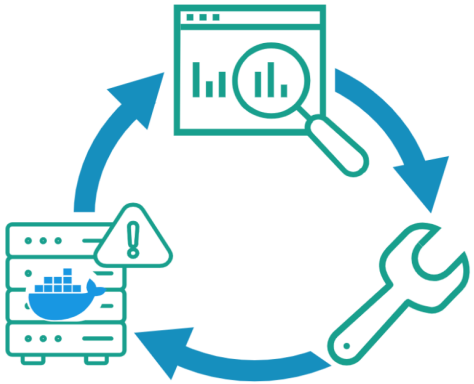
Docker client commands

```
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
```



Use cases

- Development, testing or DevOps
- Troubleshooting
- Demonstrations
- Shared environments
- Resource contention
- No installation \ patching



Introduction to Kubernetes

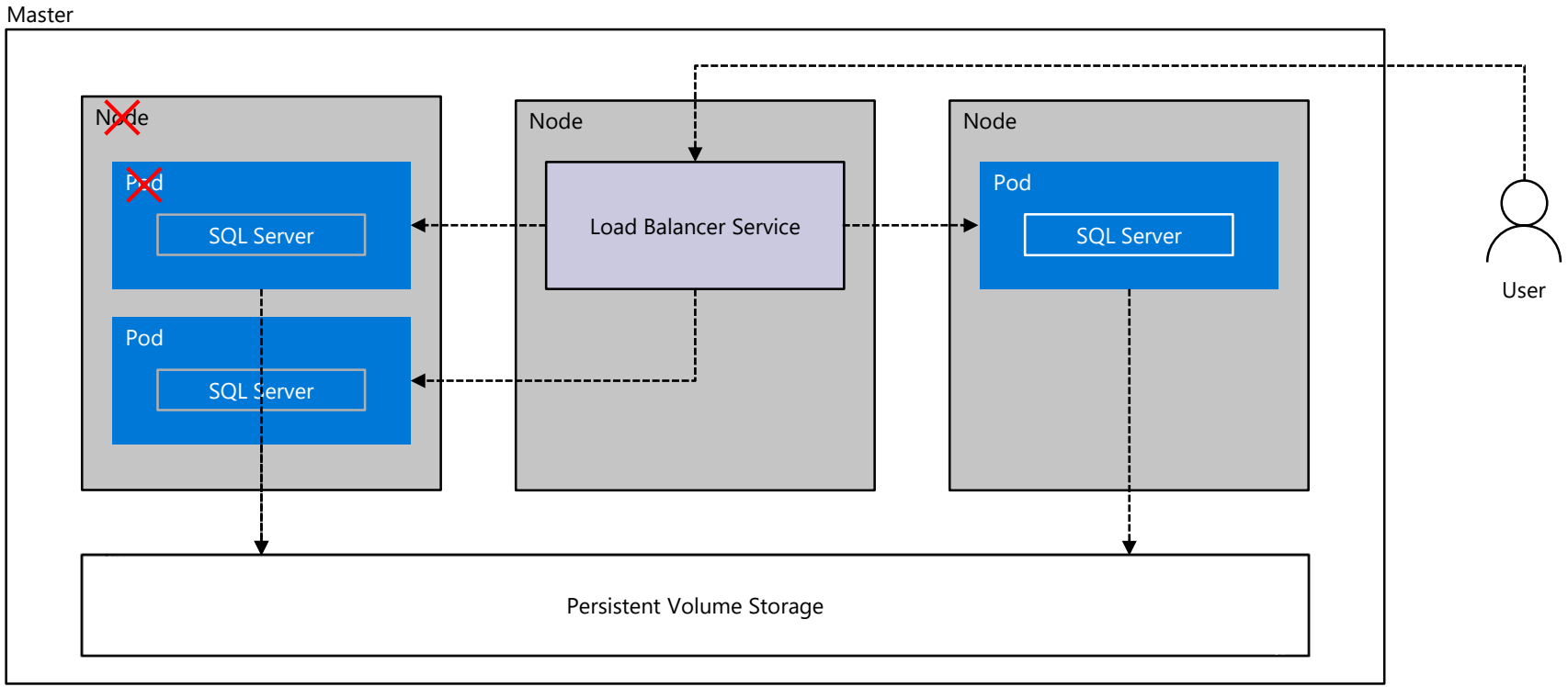
- From Kubernetes docs:

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

Kubernetes architecture

- Masters
  - Multiple moving parts
  - Runs on a single server
  - Tells what to do
- Nodes
  - Do the work
  - Aka “minions”
  - Reports the state backup to the master

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



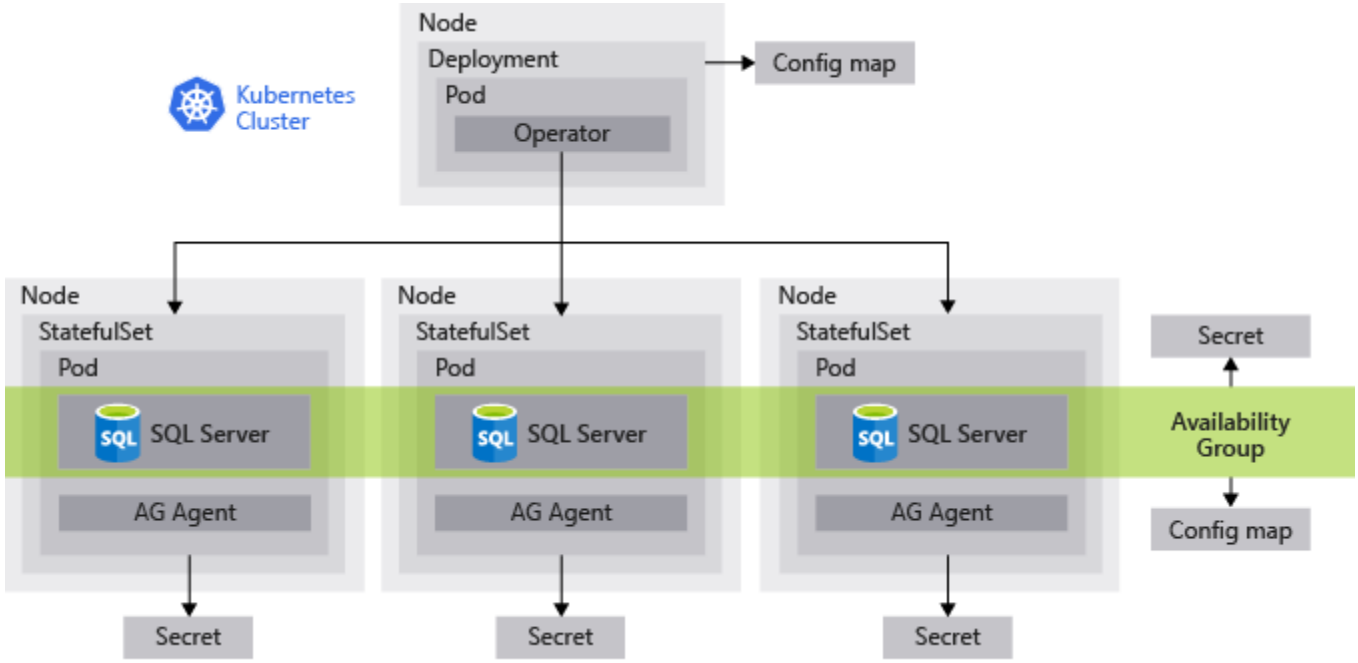
24HOURS OF PASS

Advantages

- Easy to use – Declarative configuration
- Self healing – Built in HA
- Platform agnostic
- Compute and storage layer are separate
- Load balancing
- CI\CD

Demo

Microsoft intel



24HOURS OF PASS

24HOURS OF PASS

Thank you for attending

Learn more from Carlos Robles

@dbamastery

crobles@dbamastery.com

@sqlpass
#sqlpass

@PASScommunity

Microsoft intel