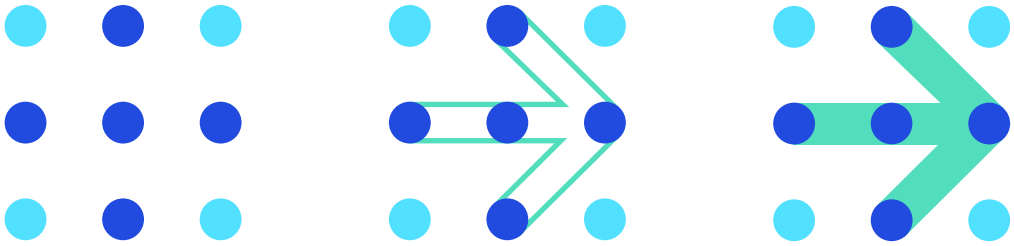
WHITE PAPER



Optimizing Data Management in Containers with Kubernetes and DataStax

DATASTAX

Yearly surveys conducted by Datadog and RightScale reveal that approximately 50% of enterprise companies have widely adopted Docker containers (Figure 1). Half of those production adopters are also using container / cloud orchestration software like Kubernetes, Mesos, Google's GKE, Microsoft's AKS, or Amazon's EKS. An additional 30% of enterprises are running their first production workloads in containers (Figure 1), and if keeping with past trends, they'll increase their container use by approximately 7x within the first 10 months (Figure 2).

Docker Adoption Status by Infrastructure Size

Figure 1. Usage rates increase with infrastructure size

Usage Among Adopters

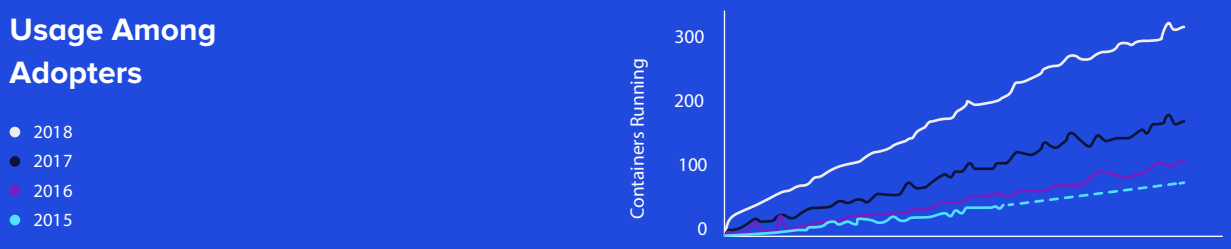


Figure 2. Docker deployment has increased 75% in one year

Why are these technologies seeing so much momentum right now?

Containers initially gained popularity for their ability to package applications in a more efficient way than was possible with virtual machines. But other benefits have now been identified, such as the ability to move workloads between different platforms, deliver high utilization of compute resources, allow disparate development teams to integrate various DevOps environments, improve speed of delivery, more quickly deploy applications with identical configurations and resource settings, and deliver simplified lifecycle management.

Orchestration technologies emerged as a way to manage these containerized applications throughout the data center. Kubernetes is the leader among many other container orchestration platforms. Kubernetes offers cloud-native tools and makes it easy to deploy and scale stateless applications in hybrid and multi-cloud environments with Kubernetes Operators that introduce abstraction for non-trivial application deployments and wraps logic for provisioning and lifecycle management.

The logical next step for these technologies is to do for databases what they've done for applications. But there are a few obstacles standing in their way:

- **Containers are immutable.** Bit by bit, they do not change. Databases have configuration files with settings that are like tunable knobs.
- **Containers should not store state.** State = Data. Databases store data that must outlive the container and thus be stored outside of the container.
- **Orchestration technologies often expect uniform clusters of nodes that all work the same.** Most databases have masters, slaves, resource managers, and other complex hierarchies of node types that drastically complicate orchestration options.
- **Orchestration technologies are used to dealing with containers that are all independent of each other.** Assigning new IP addresses or namespaces works because stateless nodes don't communicate. Databases need to be aware of their neighbors. For databases with node hierarchies that have to keep track of worker bees and their functions, this is especially difficult.

Let's look at some options for overcoming these issues, and specifically how DataStax Enterprise (DSE) makes things exceptionally easy.

01

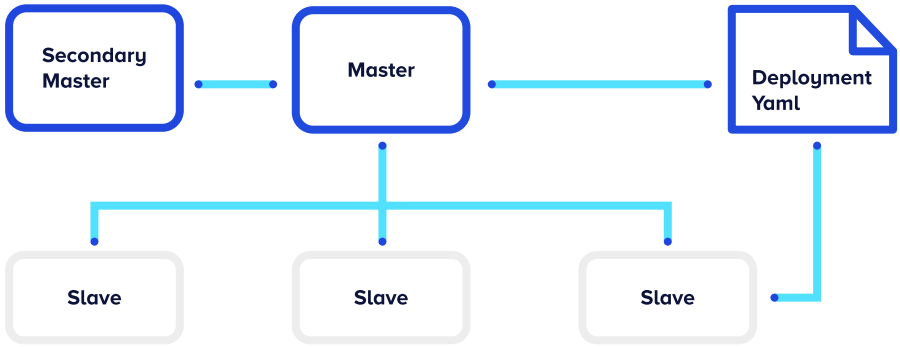
Containers and Database Architecture

The underlying architecture of a database plays a big role in how hard or easy it is to use containerization. In practicality, there are really just two designs to compare: those with hierarchies termed master/slave, and those without, which are peer to peer and termed masterless.

Architectures

First, let's look at how master/slave architectures work and how they increase the complexity of containerization and break the design principles of orchestration technologies like Kubernetes.

Master/Slave Architecture Kubernetes Deployment



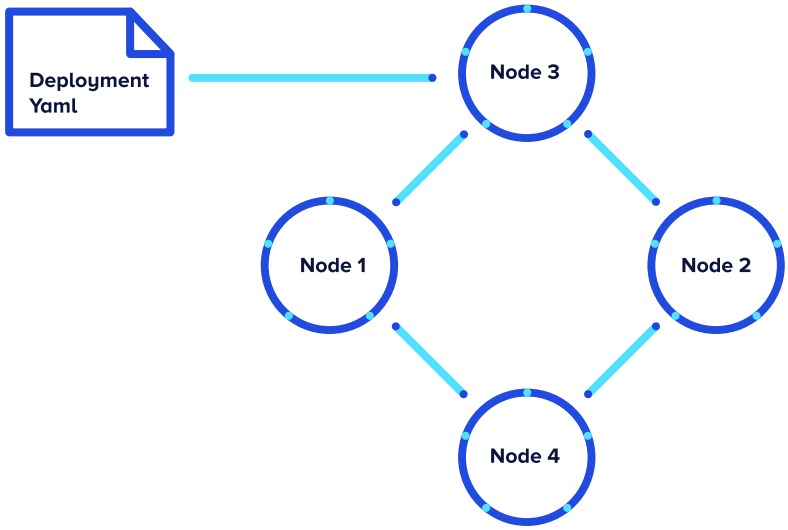
This model necessitates the use of multiple Docker images. It also requires additional resources (pods) for the master or even more resources if you want to deploy a secondary master for failover purposes. The tooling you build will also be more complex because it has to take into account the interaction between the containers/pods.

Building this complex tooling in Kubernetes hits on the rough edges of a still maturing technology. The increased risks can quickly negate expected value.

Masterless Architecture

In contrast, DSE, which is based on Apache Cassandra™, was designed to solve problems of scale and replication by embracing a masterless architecture that scales linearly on commodity servers. In a masterless design, all nodes are equal (think one download installed the exact same way on all nodes). The application can read or write to any node in the cluster and no single point of failure exists.

Masterless Architecture Kubernetes Deployment



In a masterless model, you only need one Docker image versus many and only need to maintain a single set of Kubernetes tooling to run the database. This makes development and deployment with containers and orchestration dramatically easier than databases with multiple node types.

This makes development and deployment with containers and orchestration dramatically easier than databases with multiple node types.

02

Containerization and Orchestration with DSE

With this approach, we've crunched 15+ hours of total testing time into 20 to 60 minute testing rounds. The result is that our developers get feedback much faster.

Containers

Because containers have now gone far beyond the early adopter phase, we take supporting them with DSE very seriously, as evidenced by:

- DataStax offering production support for customers running DSE in customer-built containers since September of 2015.
- In 2017, we released non-production Docker images for DSE, DSE OpsCenter, and DataStax Studio.
- After significant testing and more than one million downloads on Docker Hub, we released them for production use.

We understand why our enterprise customers are adopting container-first strategies, because we're doing the same. At DataStax, we use Docker for many of our unit, continuous integration, and functional tests. Doing so enables our Test Engineering team to run many tests in parallel on a single machine and/or test cluster. With this approach, we've crunched 15+ hours of total testing times into 20- to 60-minute testing rounds. The result is that our developers get feedback much faster.

At the corporate level, you can imagine the cost savings of running tests in one-fifteenth the amount of time with 5x parallelization. We want our customers to have this same experience! That's why we've made the same Docker images we use available on Docker Hub. Just head to hub.docker.com and search for DataStax.

Orchestration

Much of the application side of DataStax Managed Cloud is backed by Kubernetes, and we're seeing a similar trend with customers using operators to manage stateless applications in production, which need to scale up and down in number based on demand. These customers often use an Open Service Broker to connect Kubernetes-managed applications to a DSE cluster.

Although we have customers deploying and managing DSE using Kubernetes in production, we do not yet recommend everyone do so or provide production-grade tooling. DataStax offers a non-production Kubernetes Operator for DSE, you can find more information about it on our Kubernetes page.

03

Recommendations

Running containerization and orchestration technologies with DSE is much easier than with other database management systems that are not masterless in design.

Based on our own internal experience as well as that of our customers, we offer the following recommendations:

- You can confidently run DSE, DSE OpsCenter, DataStax Distribution of Apache Cassandra, and DataStax Studio in containers.
- Simply manage an external DSE cluster from Kubernetes using an Open Service Broker or Kubernetes Operator for DSE.
- Consider Kubernetes for test and development environments until you are confident in your team's ability to manage DSE within that environment.
- Be cautious of using containers or orchestrators with any database that is limited in its ability to scale or that doesn't have a peer-to-peer architecture.
- Check out the DataStax Docker images now at the Docker Hub, or try a Docker-tagged tutorial from our developer blog.
- Visit the DataStax website to learn more about our Kubernetes offerings for DataStax products.

DataStax helps companies compete in a rapidly changing world where expectations are high and new innovations happen daily. DataStax is an experienced partner in on-premises, hybrid, and multi-cloud deployments and offers a suite of distributed data management products and cloud services. We make it easy for enterprises to deliver killer apps that crush the competition.

More than 400 of the world's leading enterprises including Capital One, Cisco, Comcast, Delta Airlines, eBay, Macy's, McDonald's, Safeway, Sony, and Walmart use DataStax to build modern applications that can be deployed across any cloud. For more information, visit www.DataStax.com and follow us on Twitter @DataStax.

Learn More

datastax.com

DATASTAX