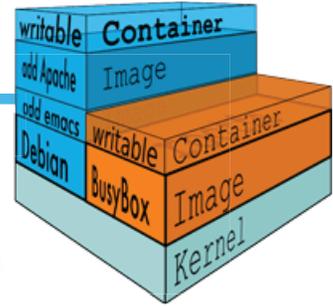


Docker + Aerospike

From Development to Deployment with Docker & Aerospike



Why Containers?

Containers are seen as a simple way to:

- Encapsulate the dependencies for the process you want to run, e.g., the packages required, the frameworks that need to be present, the code you want to execute, etc.
- Provides isolation at runtime, enabling containers (each with different dependencies, O/S Kernel version, etc.) to co-exist on the same physical host without the bloat of traditional Virtualization solutions.

Application architectures are also evolving, especially with the adoption of microservice like architectures. Your application is no longer one humongous, static binary or set of packages; it starts to look like a series of discrete services that are brought together dynamically at runtime. This is a natural fit for containers. But how does this mesh with services that require persistence or long running processes?

Why Aerospike and Docker?

Aerospike is naturally suited to container deployment thanks to attributes such as:

- Shared-nothing architecture: there is no reliance on a single node for configuration, state management, etc.
- Automatic hashing of keys across the cluster (using the RIPEMD-160 collision-free algorithm) with Smart Partitions™
- Automatic healing of the cluster as nodes enter and leave the cluster, including automatic rebalancing of the data
- Automatic discovery of the cluster topology with Aerospike Smart Client™ for all the populate languages and frameworks
- Automatic replication of data across nodes with a customizable replication factor

Key Differentiators

Simple deployment

Aerospike's integrated Docker solution provides a simple way to deploy containers to support scalable database persistence.

Elastic Scaling

With automated cluster formation and self-healing as the cluster topology changes, combined with Aerospike Smart Clients™ eliminates the need for manual reconfiguration of the application.