CASE STUDY:

For Lufthansa Systems, Customizing In-Flight Connectivity and Entertainment Systems for Its Award-winning BoardConnect Solution Depends on Stateful Containers

Portworx is the Key to Enabling Data Persistence in a Containerized Environment, and for Helping Lufthansa Systems' Partners Deliver Customized Services

Lufthansa Systems portworx

Problem

Spawning new CMS environments for hundreds of instances, and moving them freely within clusters, requires container persistence, which for Lufthansa Systems was exceedingly difficult and time-intensive to manage.

Solution

Deploy Portworx PX-Enterprise to solve the "Pandora's Box" problem of making container persistence transparent and creating stateful applications easily.

Progress

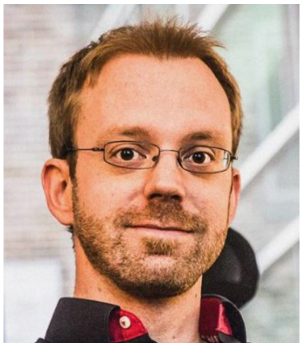
Lufthansa Systems is now equipped to spin up new CMS environments in minutes, supporting rapid customization of BoardConnect for its airline customers.

The name Lufthansa signifies the excellence and reputation of Europe's largest airline. And the name Lufthansa Systems delivers the same prestige, as the subsidiary company is one of the world's leading providers of IT services in the airline industry. The company offers its more than 300 airline customers an extensive range of successful and, in many cases, market-leading products covering all of an airline's business processes – in the cockpit, in the cabin, and on the ground.

Lufthansa Systems' BoardConnect product powers in-flight entertainment, infotainment, and connectivity for millions of airline passengers – in their seats and on their mobile devices.

Supporting BoardConnect is a framework of technology requiring flawless orchestration and a reliance on containers. According to software architect Michael Wilmes, Lufthansa Systems' IT infrastructure must match customer demands for scalability, robustness, modularity, and ease of use – with Docker containers playing a leading role.

"In order to give our airline partners the flexibility they need to bring in their own custom corporate brand, seamless travel experience, and custom services, our IT infrastructure must meet or exceed those demands."



Michael Wilmes
Software Architect
Lufthansa Systems

Persistence Pays

"Increasingly, we found ourselves building our infrastructure around microservices and containerization, and bridging development and operations – the organizing principle of DevOps," said Wilmes. "And we saw these goals as achievable, even as our development iterations and customizations picked up in volume and speed."

"Still, as we saw benefits materialize, we identified a missing link – data persistence – and we found ourselves hoping that data persistence would 'catch up' to the flexibility, ease of use, and speed that we needed to achieve in a containerized environment."

Given the critical importance of persistence, Lufthansa Systems found Portworx PX-Enterprise to be a "natural fit" for its demands. "It is free of cloud, use-case, and vendor constraints," said Wilmes. "It's fast and transparent for traditional, cloud native, and third-party applications."

BoardConnect is designed with microservices in mind, and it runs in Docker Swarm environments with Consul-backed service discovery. While data exchange with BoardConnect typically leverages clouds' object storage features, Lufthansa Systems maintains a large number of services that persist to regular block storage.

Lufthansa Systems
BoardConnect

But it's Lufthansa Systems' Content Management System (CMS) that allows BoardConnect to be customized to suit the needs of the company's diverse airline customers. CMSs, by their nature, follow a traditional persistence paradigm, such as saving data to disk or to a database, said Wilmes. When IT wants to auto-instantiate those CMS environments on developer-committed projects, or at a later stage wants to manage customers' individual CMS production instances, the need for proper Docker orchestration surfaces.

"Container-based development allows us to speed up the overall development lifecycle," said Wilmes. "In addition, it allows us to harmonize the orchestration of applications –their deployment and management – as the number of applications increases, which occurs naturally and fluidly in following a microservices-oriented approach."

Orchestration, he said, is within reach when his team can achieve flexibility in container persistence. “With Portworx, we can now also spawn full-blown CMS environments in a matter of minutes – versus hours in the past – without requiring manual intervention. For production, it further allows our operations to move CMS environments around in the cluster, and the corresponding data ‘follows along’ transparently.”

The State of Things

“Having a tool that caters to all deployment scenarios – from clouds to OEM hardware – is of tremendous value to us,” said Wilmes. “Portworx extends the coverage of our Docker-based approach to accommodate stateful and stateless applications equally well, allowing us to apply the same principles of lifecycle management throughout.”

“With our stateless applications, the benefits of container-oriented development and deployment came about easily and instantaneously. The next logical question was how to enable those benefits for applications that persist data in a more ‘traditional way.’”

This challenge takes different forms, he said. Some services require high-performance I/O using block devices. Other services, such as CMS products and databases, might be sourced from the community and may not natively adopt cloud storage. He added that Portworx’ support for stateful applications ensured its viability for the purpose.

But Operations had a challenge of its own with Docker – which Wilmes discovered could be overcome by adopting Portworx.

“When we started with our early host mounts, and later Docker-named volumes, we could already speed up the development lifecycle of applications. However, in Operations, those containers were effectively tied to their Docker hosts, where they would find their data on subsequent runs. That impaired cluster manageability and container availability. While other containers could freely ‘roam’ around the cluster, those persisting containers would require extra – often manual – care if ‘requested’ to move to a different cluster node.”

Wilmes said that some container persistence solutions are geared towards bridging storage appliances and Docker, but that they would require his team to factor in 1) storage or cloud vendor dependencies, which would contradict the desired universal applicability of a container persistence solution they were striving for, or 2) the type of storage the platform would manage. Those two conditions ruled out adopting other solutions. Furthermore, he said, his team needed a solution that would add the least number of additional "moving parts" to the infrastructure stack.

"These further challenges reinforced our decision in favor of Portworx PX-Enterprise. Even in its first release, the Portworx solution looked like the cleanest Docker-like approach for dealing with the issues at hand, while fitting the economic needs of both small and large deployments. This recognition evolved to adoption early on in our trial of Portworx, during which we found the company to be very responsive in its support."

Deploying Portworx

Wilmes said that his team found Portworx to be "well documented" right from the start. "As all required components for PX-Enterprise are deployed as containers themselves, setup was faster than expected – faster, more straightforward, and leaner than what we had experienced when we tested Flocker and ScaleIO."

"With some solutions, we felt like we were opening a Pandora's Box, despite the fact that we were simply trying to solve transparent container persistence. Portworx containers allow our Docker hosts to remain virtually clutter-free – that is, free of additional host-installed software stacks. Portworx has fewer moving parts, less to understand, and fewer things to break.'

Today, he said, his team can treat all type of containers in the same way. The team benefits from containerization on all fronts: from cloud-native microservices, to traditional CMS systems, to databases. "Being able to auto-launch instances of arbitrary development branches – including full CMS systems, their databases and filesystems – at any given time really makes a difference." Being able to use the identical approach to Docker persistence with a range of cloud providers and in a physical datacenter was a welcome addition, he added.

"For everyday operations the Portworx solution operates transparently in the background, and administrative tasks are easy to execute through the provided Lighthouse UI, or CLI (Command Line Interface) when necessary. Upgrades are straightforward, and metrics from the storage subsystem are available in a standard JSON format."

"Once Portworx was set up in the Docker cluster, the overhead of managing stateful containers became virtually zero," he added. Back to the time when his team began using Docker, deployments would require preceding efforts as a

"Our requirements were stringent: neat integraton into available Docker orchestrators, cloud vendor-agnostic and storage-agnostic setup, near-native block-level performance, capability for on-premises deployment, and ease of setup and maintainability. Portworx saw us through this maze."

means of identifying, for example, a location on the network with sufficient host-based capacity available. In addition, his team would likely need to pin those containers to their hosts for the foreseeable future in order to ensure that the hosts could find their data again on restarts or reschedules.

"Those days are over," he said. "What's more, Portworx cuts the time-to-deployment of our stateful containers to what we've come to expect with stateless containers."

What's Next?

Currently, Lufthansa Systems' Portworx roll-out centers on BoardConnect's product infrastructure, which includes test, integration, and production environments. "But we are certainly not the only department with a certain percentage of containers that need to persist locally and transparently," said Wilmes.

"We feel confident in Portworx's roadmap and fulfillment in supplying what we think are the most important missing pieces of the Docker ecosystem, ultimately making Portworx PX-Enterprise our road to running containers in production."

Learn More

www.portworx.com
www.lhsystems.com

portworx

Portworx, Inc.
4940 El Camino Real, Suite 200
Los Altos, CA 94022
info@portworx.com | www.portworx.com