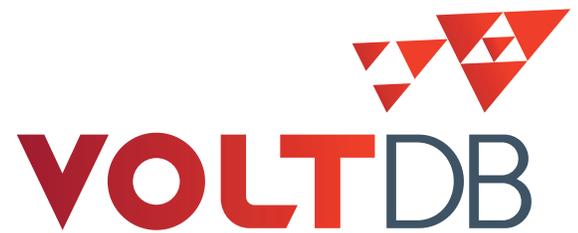


SMART DATA FAST.™



ENABLING MICROSERVICES IN THE CLOUD WITH VOLTDB, AWS, AND DOCKER

CASE STUDY

INCREASING AGILITY BY DEPLOYING MICROSERVICES

[Full 360](#) is a New York-based consulting and systems integration firm with deep experience in analytics, performance management, and business intelligence as well as data warehousing systems architecture, design, and implementation. A major airline operator engaged Full 360 to develop a managed service to support a more agile customer loyalty program that would allow the airline to launch campaigns in days rather than months. Full 360 developed an innovative fast data solution that leverages cloud-based microservices to build smarter, more flexible infrastructure to support customer engagement and loyalty campaigns.

Microservice architectures rely on developing applications as a suite of small services, each running its own processes. A microservice architecture puts each element of functionality into a separate service and scales by replicating these services, allowing microservices to leverage infrastructure automation to support continuous delivery. In a microservice architecture, complex applications are made up of small, independent processes communicating with each other using language-agnostic APIs, with each service focused on one specific task. Full 360 built this modern, scalable, responsive infrastructure using VoltDB's fast in-memory relational database, AWS cloud services, and Docker for rapid service deployment.

THE CHALLENGES: SPEED AND SCALE

Full 360 was faced with developing and managing a scalable solution that would support one of the largest customer loyalty programs in the world, which has over 10 million members. The international airline had big data infrastructure in place, but wanted to accelerate campaigns development. “The challenge was to deploy a fast data infrastructure nimbly and quickly,” said Rusty Ross, Solutions Architect for Full 360. “Historically, this loyalty program has had a very high open rate for email campaigns, so we had to prepare for very high traffic spikes almost immediately after the launch of new campaigns. That dictated the need to focus on a highly scalable solution. We’re not serving customers static data; each click-through leads to complex web applications serving robust content. Because this loyalty program is so valuable to our client—and because it’s been so successful—downtime is not an option.”

As Full 360 evaluated its architectural options, legacy approaches were quickly excluded. According to Ross, “We did not deploy a monolithic web server and a monolithic code base, nor did we deploy a legacy database like MySQL, PostgreSQL, or even a NoSQL solution. We wanted to avoid over-provisioning server infrastructure to handle traffic spikes, and we wanted to avoid the risk of under-provisioning server resources and hitting performance problems. When you’ve got a monolithic server infrastructure in production, it’s very hard to update and hard to maintain. It’s difficult to delegate development, and when you’ve got a single code base, it’s hard to develop new chunks of a campaign that hasn’t been deployed yet or to improve chunks that already exist in production. A legacy approach is slow because data resides on disk, so we needed an in-memory database that

would allow us to analyze real-time data and deliver decisions in milliseconds.”

FAST, IN-MEMORY DATABASE

Full 360 selected [VoltDB](#), the only in-memory, NewSQL solution that delivers Smart Data Fast. VoltDB offers the performance of in-memory, the scalability of NoSQL, and the transactional consistency of traditional relational databases. With VoltDB, you can build transactional, database-oriented applications against data feeds that were previously limited to stream processing methods because of scale.

“I’d say first and foremost we selected VoltDB because it’s fast,” Ross stated. “Because the data resides in memory, we’re not disk-bound. In this implementation, velocity is of paramount importance. We’re serving lots of small transactions to many, many members and we just can’t tolerate downtime. Beyond that, unlike a typical MySQL or Postgres installation, VoltDB is easily horizontally scalable. As our traffic needs increase over time, we can easily scale up the capacity of our VoltDB clusters and process even more transactions per second. VoltDB is also ACID compliant, which is critical in an application like this.”

Full 360 was experienced in deploying VoltDB, having previously implemented it in several other accounts. “VoltDB is a long-time partner of ours and we trust them,” Ross said. “We’ve deployed VoltDB in a number of situations and it’s served us well. We did not have to look far to choose VoltDB because this deployment required high availability, scalability, ease of migration, and speed of migration. We felt that VoltDB was a particularly good fit for this loyalty program. We’re glad we chose it and we’re not looking back.”

DEPLOYING MICROSERVICES ON DOCKER CONTAINERS

Full 360 architected an innovative solution based on deploying microservices in [Docker](#) containers on Amazon Web Services ([AWS](#)) cloud infrastructure. Docker is an open platform for building, shipping, and running distributed applications. Consisting of Docker Engine, a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for sharing applications and automating workflows, Docker enables applications and services to be quickly assembled from components.

“We containerized VoltDB into Docker,” explained Ross. “Docker is a containerization platform which lets us pack applications and software into isolated containers and deploy them across the hosts. We were able to develop a single Dockerfile to build the containers which pulls VoltDB code. We can build everything we need within a Docker container in essentially a single command line interface step. We can get clusters up and running very quickly and easily. Because it is so easy to instantiate new clusters, we can spin up a parallel cluster in production while our first production cluster is running. If, for instance, we need to do a major schema migration or other migration-related activities, we can spin up a parallel cluster and instantiate it from a snapshot. When it’s up and in production we can do a fairly major, seamless migration with no downtime.” The following are examples of some of the microservices, each deployed in Docker containers:

- Cross-promotion Campaign
- Rewards Points Calculator
- Status API
- Goals Campaign
- Member Data API

The Docker containers also scale efficiently. “VoltDB is highly scalable, and we’ve found that to be even more the case in a Docker environment,” Ross said. “For instance, if we start with a three-container cluster but our capacity needs increase, it’s very easy for us to add additional containers to that cluster to handle more capacity. We’ve found Docker and VoltDB to be fantastic fits in this solution.”

By avoiding the monolithic approach of building a single application, Full 360 maintains greater flexibility for upgrading and enhancing each of the microservices. According to Ross, “We did not work from a monolithic code base. Instead, we considered the concept of a fleet of web services. We broke that proverbial monolithic code base up into distinct microservices. Working within the microservices model, we found immediate advantages. The first advantage is that it allows individual microservices to be horizontally scalable. For instance, if a particular campaign has unusually high traffic demands, we can horizontally scale that microservice independent of the other services. Another benefit of working within the microservices model is the ease of software updates.”

If Full 360 deploys a new release of a microservice, it can be released into production alongside instances of the older version. “When the newer version is available in production, we can retire older versions,” Ross said. “This allows us to do a live version update right in the midst of production with no downtime. And in doing that, we haven’t touched any of the other microservices because all microservices are decoupled from one another.”

The microservices allow Full 360 to delegate development power. Ross explained, “We can have an isolated development team work on the Status API while we’ve got an entirely independent team building the next microservice of a campaign that’s

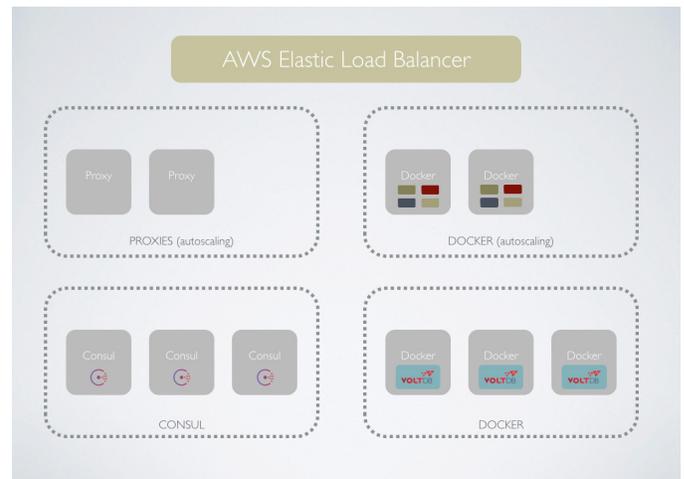
going to launch in a couple of weeks. It's a great benefit in an infrastructure like this because we have very fast deployment demands. We've found that the microservices model has facilitated agility and it's also made the client very happy."

Balancing the traffic load across microservices and managing dependencies between microservices requires efficient service discovery. Full 360 deployed the open source [Consul](#) tool for allowing microservices to register themselves to be located via a DNS interface. "Each container registers with Consul, which institutes a health check on those registrations every 10 seconds," said Ross. "If a container fails to respond to the health check, Consul will know that health check has failed almost instantly and will not direct traffic to that container."

INTEGRATING WITH CLOUD INFRASTRUCTURE

Full 360 architected the solution to take advantage of the elasticity and scalability of cloud infrastructure. "I consider the bedrock of this approach to be the database, and VoltDB is giving us the performance and scale we need," said Ross. "We've containerized all of our production resources in Docker and that provides us with major scalability, availability, and deployment advantages, and we've solved our discovery challenges in an elegant way using Consul. We built all this infrastructure in the cloud using AWS, which allows us to take advantage of automatically scaling when needed to meet peak traffic demands."

The solution leverages AWS Elastic Load Balancing to automatically distribute incoming application traffic. It enables Full 360's airline customer to achieve greater levels of fault tolerance by seamlessly providing the required amount of load balancing capacity needed to distribute application traffic.



VoltDB is deployed in Docker containers to enable flexible and scalable real-time microservices

Additionally, Elastic Load Balancing offers integration with AWS Auto Scaling to ensure back-end capacity to meet varying levels of traffic levels without requiring manual intervention.

Proxy nodes register with AWS Load Balancing, which directs inbound requests to them. They inspect each request and then forward it to a microservice on a Docker container. The Consul nodes provide discovery services for containers and conduct the health checks so the proxy nodes can forward requests to healthy microservices. For example, if a browser requests access to the Member Data API microservice, Elastic Load Balancing will send the request to a proxy node, which will perform a DNS query to Consul to identify an appropriate container hosting the Member Data API microservice.

Once the proxy node knows where a healthy instance of a container with the Member Data API microservice is located, it will direct that request to it and the microservice will run the browser query against the member database that's being served by VoltDB. Since each microservice performs a single specific function, several microservices can be linked together to complete complex tasks.

Full 360 successfully designed and implemented this fast data solution for one of the largest and most successful loyalty programs in the world, and the number of users and the volume of transactions continue to grow. The international airline now relies on the latest database and cloud technologies to stay ahead of the competition and turn data into deployed services quickly and easily, enabling them to respond rapidly to changing needs and insights. Ross concluded, "VoltDB is performing like a charm, and we're seeing hundreds of thousands of successful transactions per second."