portworx REACHFORCE

# Case Study: ReachForce Containerizes 200+ AWS Instances with Portworx to Reduce Data Center Footprint and CPU Utilization

## CHALLENGES

- AWS EBS storage severely under-utilized with only 25% percent of purchased capacity being used
- At the same time, EC2 instances were over-provisioned by 100%, increasing the cost of operating the ReachForce SaaS platform
- Data services backing the ReachForce SaaS platform must be performant, available and protected

## SOLUTION

- Docker-based container management platform for a scalable, easy-to-use container-as-a-service offering and hosted web applications
- Portworx Enterprise for cloud native storage and data management

## RESULT

- Portworx aggregates all the EBS storage on the container hosts and thanks to thin provisioning, ReachForce was able to serve the same number of volumes and the same amount of data with half as much AWS storage
- Containerization, including mission-critical data services, allowed ReachForce to reduce their AWS compute footprint by 50%
- Because Portworx handles the connectivity between the containers and data, ReachForce's container placement strategies can be quite simple. It is not necessary to ensure that the container is running on the same container host where the persistent volume exists in order to ensure application performance and availability

**We spoke to Tom Watson, DevOps Engineer at ReachForce, a marketing automation company from Austin, TX, about building Dockerized applications on Amazon using ECS. This is a transcript of that conversation.**

**Tom Watson**
DevOps Engineer at ReachForce

## KEY TECHNOLOGIES DISCUSSED

Data Center – AWS
Container Runtime – Docker
Scheduler – ECS
Stateful Services – MongoDB, etcd, Jenkins

## Can you tell us a little about what ReachForce does?

ReachForce is in the marketing automation space. We make marketing leads more valuable and actionable. Most leads are limited to business card information: first name, last name, title, company, street address, email address, and phone number. They flow into marketing automation systems (Marketo, Eloqua, etc.) from a variety of sources. ReachForce is partnered with Marketo, Eloqua, and Salesforce, and our service can validate, deduplicate and enrich them with many more fields of information about the business. A salesperson can be a lot more effective if the lead has an organizational structure, annual sales, accurate job titles, etc. And with this level of detail, sales automation can be much more effective (e.g. automating distribution of leads to salespeople and regions).

## What is your role at ReachForce?

My title is Senior DevOps engineer. I'm responsible for the ReachForce data center and have been developing a plan to containerize our 200+ AWS instances. This has involved building the Docker infrastructure (ECS, etcd, confd) and the CI/CD pipeline (Git, Jenkins).

## How are you using containers at ReachForce?

We currently have an AWS data center comprised of four environments: Dev, QA, Performance Lab, and Production. These environments comprise about 215 EC2 instances.

About three years ago, we re-implemented our SaaS offering in a micro-services architecture. At the time, industry best practice was to use a separate (and appropriately sized) AWS instance for each micro-service. Over time, it became apparent that that choice had several negative consequences. We were having to maintain over 200 Linux OS instances, and our average CPU utilization was significantly below best industry practices. By that time, however, containerization was becoming mainstream and offered the promise of better resource utilization and manageability. That's when we started us down a path of re-implementing the data center as a containerized architecture. We believe that we can collapse our data center footprint from 215 EC2 instances down to as few as fifteen [using containers]. And we have set a goal of  reaching CPU utilization of 25%.

**"We believe that we can collapse our data center footprint from 215 EC2 instances down to as few as fifteen [using containers]. And we have set a goal of reaching CPU utilization of 25%."**

We stood up the first development cluster in November, and now we are about to roll out a preview of our first application based on containers. It is a customer portal and consists of a database, a two-container SSO solution and a customer web portal that reinforces the value proposition of our SaaS service. We are using etcd and confd for service discovery and environment-specific configuration. The CI/CD pipeline is built out using Jenkins, containerized and with its workspace on a Portworx volume. We elected to go with AWS ECS for container orchestration, largely for simplicity, and because we have embraced a strategy of leaning into AWS.

Now that we have a ReachForce base image (built on Tomcat7-Alpine), conventions for etcd configuration, working CI/CD pipeline, and conventions for ECS Tasks and Services, porting other micro-services (most utilize Tomcat) will be very straightforward. Once we get beyond the first few micro-services, we will begin using CloudFormation to codify everything.

## You emphasize automation being so important. How does automation factor into your decision about data management solutions?

Across the ReachForce SaaS architecture, we have a number of Mongo, Redis and RabbitMQ instances that need a persistent backing store. We also run Jenkins and our Docker registries on Docker. So, we needed a highly reliable, persistent storage. While Docker now supports volumes, we didn't see the native implementation as robust. It can't provide redundancy, performance, and manageability that we needed.

**"While Docker now supports volumes, we didn't see the native implementation as robust. It can't provide redundancy, performance, and manageability that we needed."**

Just as we were looking for a solution to these challenges, Portworx came to our attention at a local Docker user meeting. After testing it for several months, we were sold. In addition to solving the reliability and persistence challenges, it also offered storage consolidation through thin provisioning.

The icing on the cake was the Lighthouse management UI portal. Setting up and managing persistent volumes on our clusters is now as easy as setting up Tasks and Services. Because Portworx handles the connectivity between the containers and the Portworx volumes, our container placement strategies can be quite simple. It is not necessary to ensure that the container is running on the same container host where the persistent volume exists.

Another challenge was to more efficiently use our AWS storage. Just by adopting Docker, we will eliminate terabytes of storage that have been dedicated to Linux. But ignoring that, we still have tens of terabytes of data storage that is, on average, only about 25% full. Because Portworx aggregates all the storage on the container hosts and uses thin provisioning, we will be able to serve the same volumes and the same amount of data with half as much AWS storage. Just as importantly, though, it allows us to aggregate our storage tiers across each ECS cluster and only add more SSD or magnetic EBS storage as the data grows.

## What advice would you give someone seriously considering running stateful containers in production?

Plan carefully. Make an inventory of your persistent data stores and specify performance and resiliency. Consider how you'd recover if the container host that the volume resides on was inaccessible. How will you track and manage all of the data stores? How will you manage storage performance tiers? Each app is different, so you need to go through the process yourself, but when I thought through all this on the front end, the value proposition for Portworx became immediately apparent.

## LEARN MORE

Portworx is the cloud native storage company that enterprises depend on to reduce the cost and complexity of rapidly deploying containerized applications across multiple clouds and on-prem environments.

With Portworx, you can manage any database or stateful service on any infrastructure using any container scheduler. You get a single data management layer for all of your stateful services, no matter where they run. Portworx thrives in multi-cloud environments.

Contact us to find out how Portworx's unique storage solution can deliver the availability, performance, and features necessary to minimize stateful application operations costs and improve revenue growth -

https://portworx.com/request-a-demo

portworx.com
www.reachforce.com

@portwx
linkedin.com/company/portworx

portworx

4940 El Camino Real, Ste 200, Los Altos, CA 94022
Tel: 650-241-3222 | info@portworx.com | portworx.com