



» CUSTOMER CASE STUDY

# Delivering Chairside Dental with AWS and Docker

Glidewell Laboratories implemented Docker on AWS to make chairside dental manufacturing a cost-effective reality.



# Executive Summary

As a premier manufacturer of dental prosthetics, Glidewell Laboratories recognized that there was a new threat: 3D printers and companies developing software to support chair side manufacturing were taking market share.

In the process of identifying a response, Glidewell engaged Nebulaworks to assist with strategy and enablement, providing expertise to catalyze the deployment of a new suite of applications on a distributed platform, including:

- Training
- Platform Build
- DevOps Tooling & Culture

“By choosing to use docker as an image artifact and orchestrate deployment on Amazon Web Services, we were able to provide Glidewell scalability and a cost-effective platform for growth in a new market.”

**Chris Ciborowski**  
CEO & Co-Founder, Nebulaworks

## The Challenge

Glidewell produces thousands of products per day, primarily with high human interaction. New technologies, like 3D printing, have created a market allowing competitors to provide chairside production.

By eliminating manufacturing costs and reducing patient wait time, competitors are gaining market share and providing a service which customers value.

Glidewell identified this as an obstacle to growth and potential loss of customers. However, even

with significant application development expertise in-house they did not have experience with containers, running them in production, delivered in the cloud.

## The Solution

Nebulaworks began by listening to our customer and identifying business inefficiencies which were preventing growth. We quickly noticed that the developers, operations team and project managers were speaking about the same technologies differently.

Starting with education, Nebulaworks provided training and leadership discussions on DevOps best practices including application build pipelines, containerization, infrastructure automation and distributed systems and their deployment in the cloud.

After a baseline understanding of technology between the developer and operations teams was established, we began a consulting engagement to identify the requirements for an internally-facing platform to support containerized applications. This Pilot environment would support continued education of the team on the tooling that would ultimately be placed into production to support the company's chairside initiative. This was wildly successful, and provided the Glidewell management team the confidence to take the next step, a production deployment of the technology stack in the private cloud.

Nebulaworks set out, iterating on the success of the development platform to design an architecture and deploy an updated container platform in Amazon Web Services. Through the lessons learned in the training and the initial pilot, the client was able to clearly articulate their needs, and we were able to quickly address new requirements and application service.

## Choosing the Right Platform

A key component to the engagement's success was Nebulaworks' ability to iterate over design and tooling choices, and best-of-breed options to support application and service requirements. After testing, it became clear to the mutual team the initial clustering framework was not adequate to support the application containers in production. As a result, the Nebulaworks team deployed a new scheduler to handle the offending applications.

This provided the Glidewell developers the granular resource allocations they required, while supporting their commercial Docker Engine runtimes and service discovery implemented in the development RPE without delaying the project.



# Outcome

Resulting from the engagement, Glidewell was able to realize the development and delivery of their chairside application stack in AWS in less than 60 days, from initial discussion to production implementation. Containers increased the utilization of the EC2 instances providing a cost reduction of 10:1 over previous AWS deployments, while automation reduced infrastructure deployments time to under fifteen minutes.