# Setup Internet Facing Docker Server

Overview:

In order to make our applications extremely portable, we make use of open source tools and deploy applications using Docker containers. This approach allows the application to be run in virtually any data center that has deployed the industry standard Docker server. Hosting the Docker server ourselves allows for more granular security management as needed to meet government regulations.

## General Process to Move the App to Another Server:

* Setup a Docker infrastructure
* Harden the server and network to meeting compliance requirements as applicable
* Deploy the Docker containers to new infrastructure
* Make the application available on the Internet

## Docker Server Setup:

We used an Amazon based CentOS virtual machine. We used CentOS because it is most similar to Redhat, but without the license and support fees. We could just as easily have setup the application to run in an alternate cloud environment such as Azure with Ubuntu, or on any of the PaaS Docker services.

* [Setup a new Amazon CentOS 7.x instance](http://wiki.centos.org/Cloud/AWS) with 100GB disk and 16GB RAM.
* Install all updates (yum update)
* [Install anti-virus](http://www.clamav.net/about.html). We chose ClamAV for this application because it is open source and well respected. Other applications may require a commercial AV product for compliance purposes.
* [Download and install Docker](https://docs.docker.com/installation/centos/) from RPM
* [Setup certificates](https://docs.docker.com/articles/https/) and configure Docker to require them for network connections.
* Configure Docker to start automatically at boot.

## Amazon Support Setup:

The items in this section would be configured differently in other environments, but this illustrates the general steps needed.

* [Open the following firewall holes](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html) from the internal admin network to the Docker host:
  + Port 22 for SSH admin
  + Port 2376 for Docker image deployments
  + Ports 80-84 for internal website testing
* [Setup three load balancers](http://aws.amazon.com/elasticloadbalancing/) to expose the three demo Internet facing URL’s to the internal containers, and to map the internal ports to a standard external http port 80.
  + In a production environment this would be the basis for a redundant, highly available web site supported by multiple containers running on multiple hosts.
* Setup DNS to point to the load balancer URL’s.