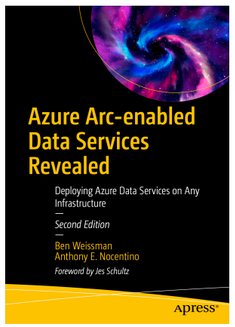
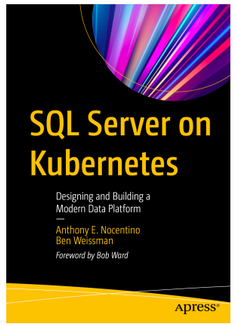
Deploying Applications in Kubernetes - Windows Edition!

Anthony E. Nocentino
anocentino@purestorage.com

Anthony E. Nocentino

- **Principal Field Solution Architect @ Pure Storage**
  - Specialize in system architecture and performance
  - Masters Computer Science
- **email**: anocentino@purestorage.com
- **Twitter**: @nocentino
- **Blog**: www.nocentino.com
- **GitHub:** https://github.com/nocentino/
- **Pluralsight Author**: www.pluralsight.com
- **Founding Organizer of EightKB -** www.eightkb.online





Agenda

- Story Time
- Building a Windows container application
- Deploying Windows containers in Kubernetes
- Rolling out Windows container updates in Kubernetes
- Compare and contrast Linux and Windows Containers
- Windows Container best practices and life lessons

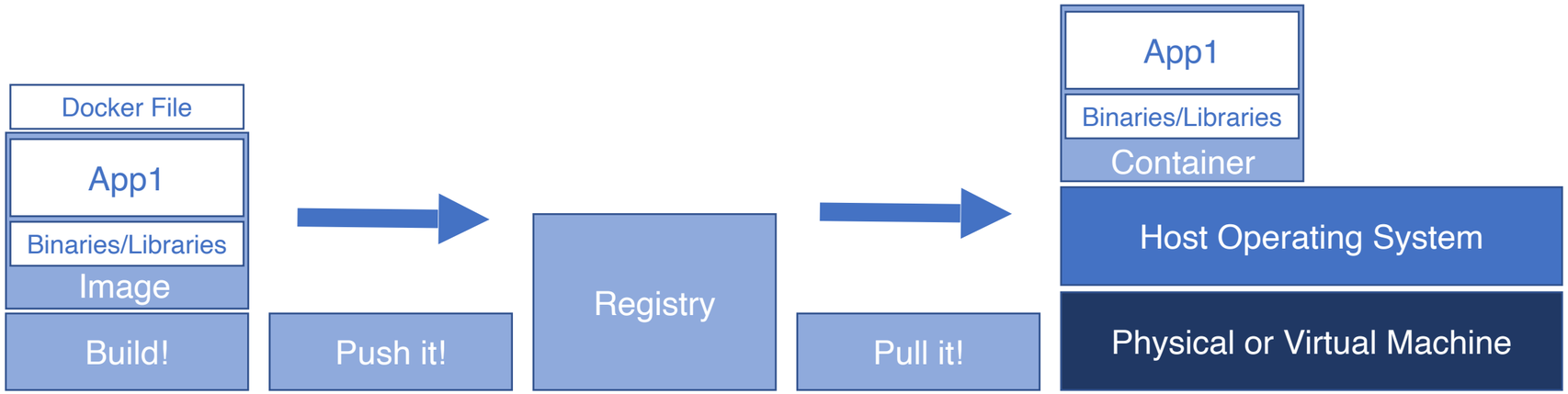
Story Time

“You use Windows containers in Kubernetes…you’re the only ones”

Anthony Nocentino - 2019

Getting and Building Containers

- **Images** – code, runtimes, libraries, environment variables
- **Registries** – where images live. Docker Hub, Azure Container Registry, internal
- **Docker Files** – defines the container image

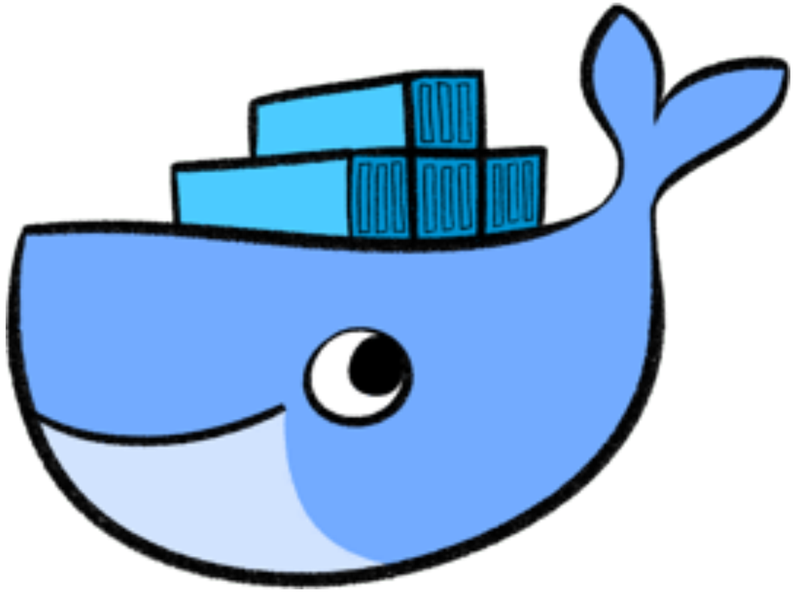


Demos

- Let’s do a lot of demos…

Building a Windows Container Application

- VB/C# Hello World IIS Web App
- Full .NET vs .NET Core
  - Should you re-platform?
- Base Image Selection
  - Nano/Core/IIS?



- **Demo: Build container images and push to ACR**

Kubernetes - Windows Architecture

- Azure Kubernetes Service
  - Linux Control Plane
- Node Pools
  - Windows Nodes
  - You still have to have a Linux node

- **Demo: Deploy an AKS Cluster with Windows Node Pool**

Deploying Windows Containers in Kubernetes

- Container pull/download times
- Container start up times
- Slow app startup
- App Pool time out
- JIT vs. Full compile
- Bad things can happen…like outages

- **Demo: Deploying a Windows container in Kubernetes**

Rolling Deploying Windows Containers in Kubernetes

- Startup and Readiness probes
- `maxSurge` to prevent over committing resources
- `maxunavailable` to keep workloads consistent
- Workload placement and Node size
- Memory Management - `Limits` and `Requests`

- **Demo: Deploying a Windows container in Kubernetes a better way**

Best Practices and Life Lessons

- Don’t have to change code…but is that a good thing long term?
- Base image complexity (Nano/Core/IIS)
- Size of containers matter
- Compile/Build times
- Container/Pod startup times
- App start up times can be long
- Deployment complexity

https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-containers/container-base-images

https://docs.microsoft.com/en-us/virtualization/windowscontainers/manage-docker/optimize-windows-dockerfile

https://docs.microsoft.com/en-us/dotnet/architecture/modernize-with-azure-containers/modernize-existing-apps-to-cloud-optimized/deploy-existing-net-apps-as-windows-containers

Best Practices and Life Lessons

- A workable solution if you know the architecture
- Set Limits and Requests
- Set Startup, Liveness and Readiness probes
- Optimizations are constantly being implemented - smaller images/faster build times
- Can Linux containers have the same challenges, yes
  - SQL Server is big and takes a minute to start

Review

- Story Time
- Building a Windows container application
- Deploying Windows containers in Kubernetes
- Rolling out Windows container updates in Kubernetes
- Compare and contrast Linux and Windows Containers
- Windows Container best practices and life lessons

Need More Data?

- **Contact Me**
  - **Email:** anocentino@purestorage.com
  - **Twitter:** @nocentino
  - **Blog - www.nocentino.com**
  - **GitHub - https://github.com/nocentino/Presentations**

- **Pluralsight**
  - Linux
  - Kubernetes
  - Azure
  - Hit me up for free access to this content