Containers - What's Next!

Anthony E. Nocentino
aen@centinosystems.com

Centino systems

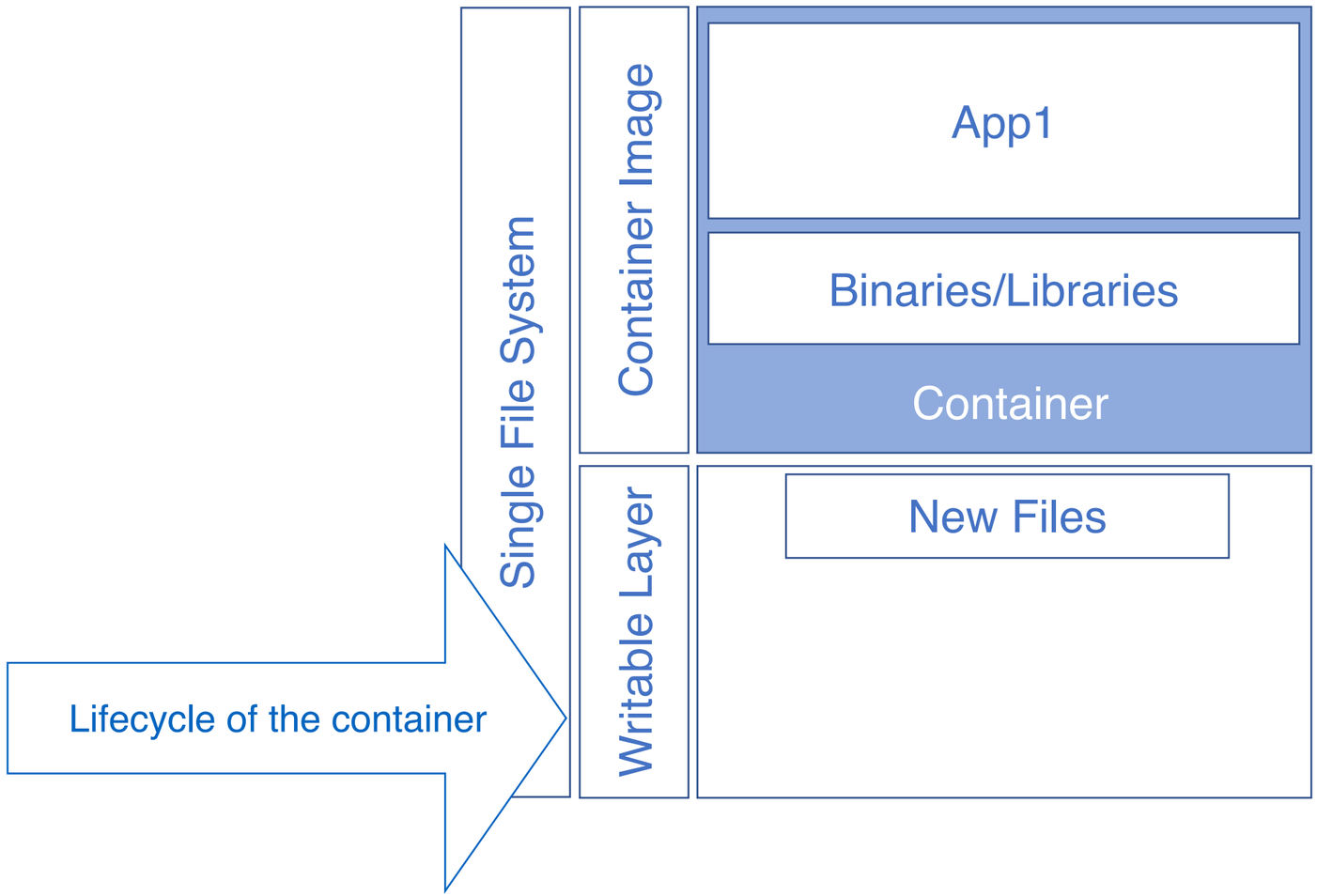
Anthony E. Nocentino

- **Consultant and Trainer**
- **Founder and President of Centino Systems**
  - Specialize in system architecture and performance
  - Masters Computer Science
  - Microsoft MVP - Data Platform - 2017 - 2020
  - Linux Foundation Certified Engineer
  - Friend of Redgate - 2015-2020
- **email**: aen@centinosystems.com
- **Twitter**: @nocentino
- **Blog**: www.centinosystems.com/blog
- **Pluralsight Author**: www.pluralsight.com

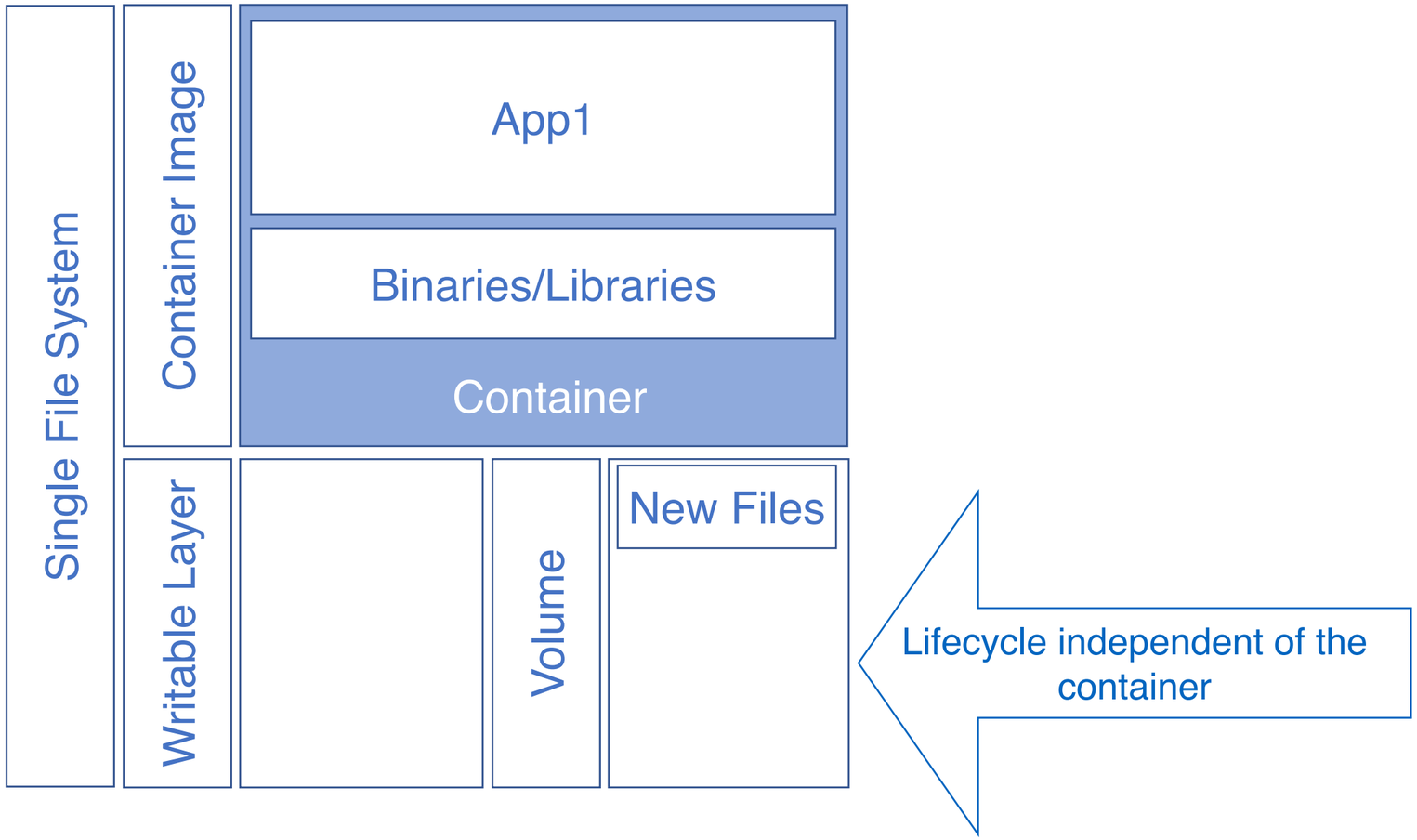
Agenda

- **Storing Persistent Data in Containers**
- **Non-root Containers**
- **Custom Container Builds with SQL Server Features and Configuration**
- **Getting Data into Your Containers**
- **Container Performance Concepts**

How Containers Store Data

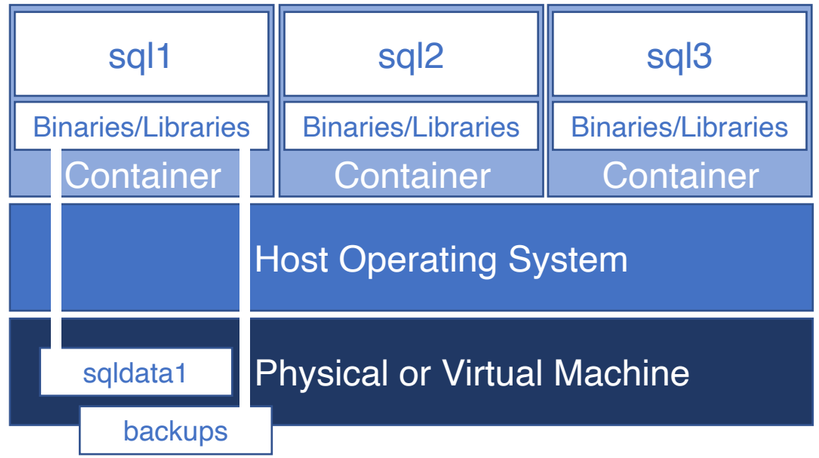


How Containers Can Store Persistent Data



Data Persistency in Containers

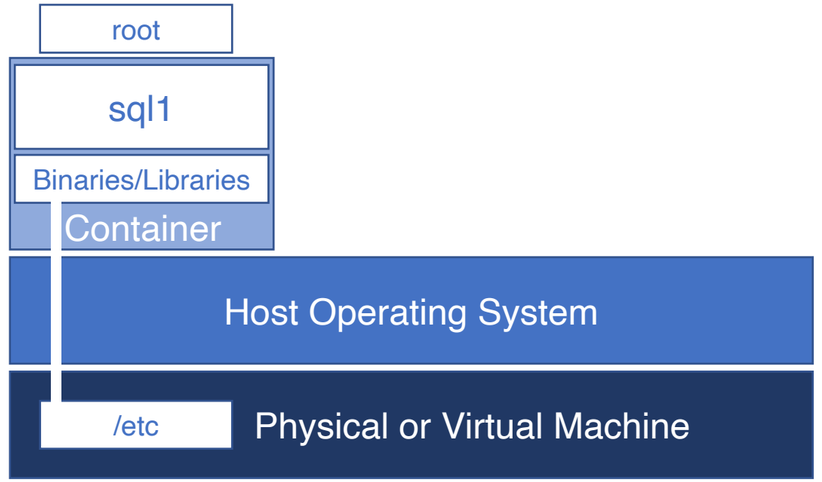
- Docker Data Volumes
  - Generally local storage on the host
  - Volume plugins enable remote storage scenarios
- Remote storage at the OS level
- You can pre-populate content
  - Backups
  - Database files
- Bind Mounts



https://docs.docker.com/storage/

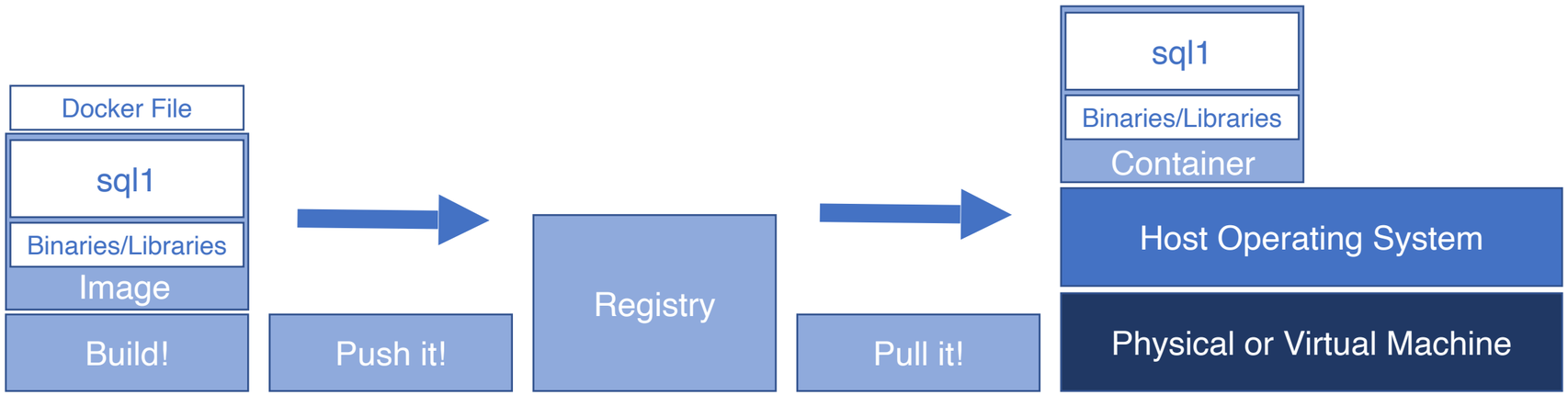
Non-Root Containers

- SQL Server previously ran as the root user
- Exposes the underlying OS to security risk
  - Docker commands are privileged
- Linux uses on UID and GID for permissions
- Now run as user mssql
- Official MS Images require no config
- When building images you'll need to run some tasks as root then switch to mssql and clean up permissions



Creating Your Own Container Images

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



Why Build Your Own Container Image?

Build Once Deploy Many

Customization

Control

Security

SQL Server Custom Container Build Process

Setup the environment	Install SQL Server	Configure SQL Server	Build the container image
Add Users	Install mssql-server	mssql-conf	Local
Add Repositories	Install mssql-tools	Environment Variables	Push to a repository
	Install Additional Packages		

https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-mssql-conf

https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-configure-environment-variables

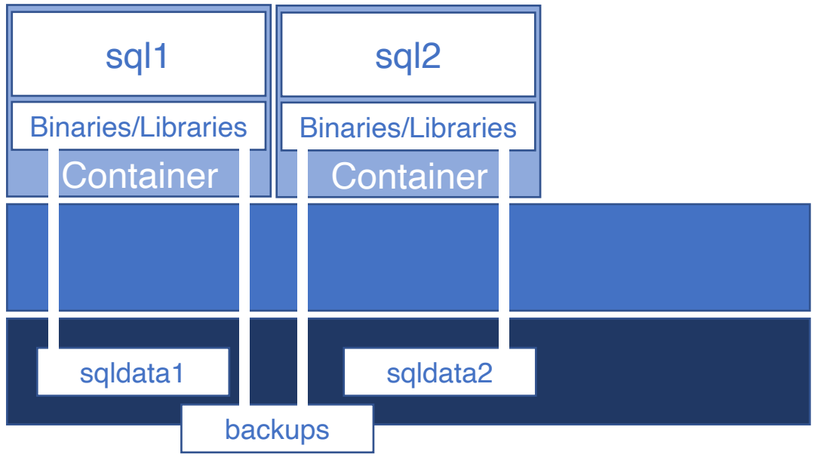
Centino systems

Demo!

- **Examine a dockerfile**
- **Creating a Custom Image**
- **Configuring SQL Server**
- **Deploy SQL Server Custom Image as a Container**

Getting Data Into Your Databases in Containers

- Should I put the databases inside the container image?
  - The size of the database is part of the image
  - On container startup, COW into the writable layer
  - Restore or attach a database on container start up
    - Manually or automatically
    - Databases need to be available to SQL Server inside the container
  - Restore or attach from a mounted volume
  - Local or remote volume
- Seeding larger databases in containers



Automatically Restoring a Database at Container Deployment

Call script to execute restore or attach

Loop sqlcmd test if SQL is online

Call Script at CMD in dockerfile

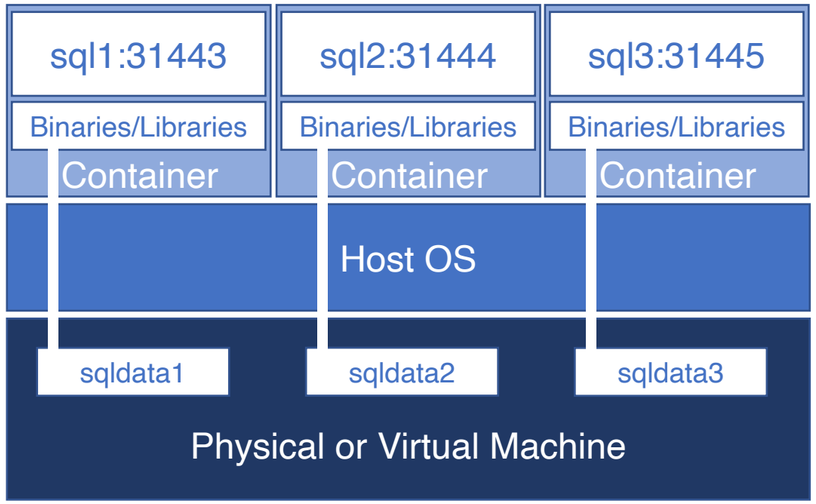
Centino systems

Demo!

- **Restoring databases inside containers**

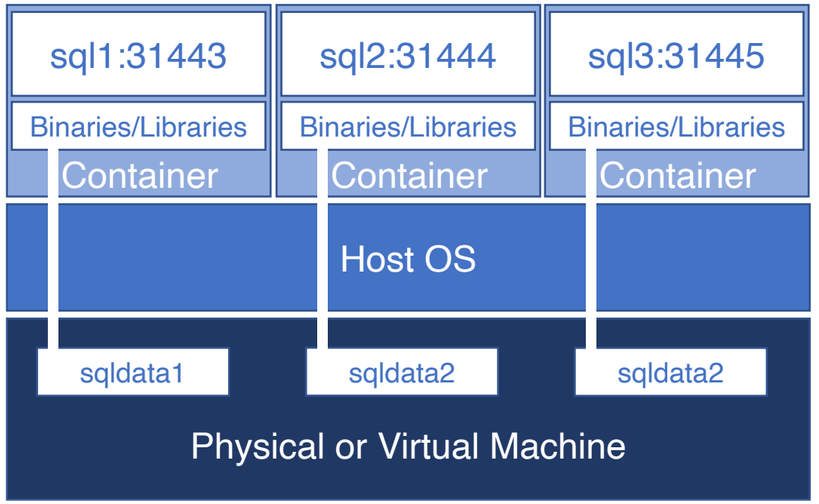
Multi-instance Scenarios for SQL Server on Linux Using Containers

- SQL Server on Linux doesn't support named instances
- Containers provide similar functionality
- Deploy with unique
  - Container Names
  - Storage for Data
  - Network ports
- Resource management is your responsibility



Container-based Performance Concepts

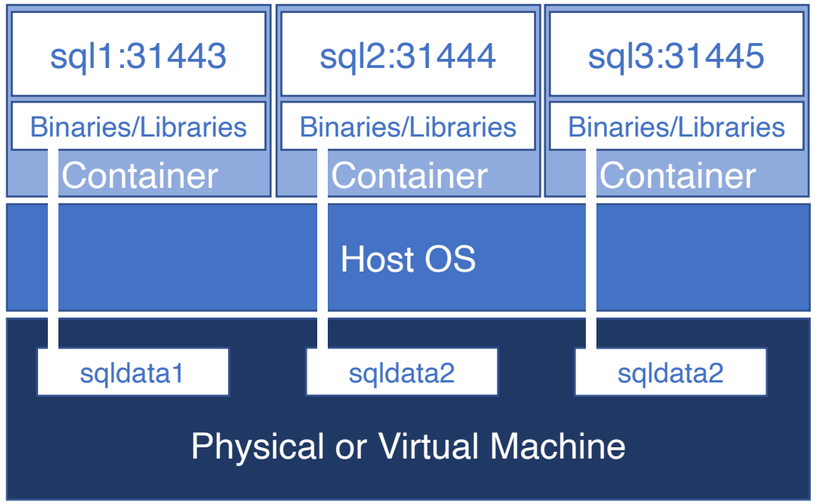
- Multi-instance scenarios
- Sharing the OS and it's hardware
- Resource controls
  - Control groups (cgroups)
- Docker allows you to control access to resources
  - CPU
  - Memory
  - Block IO
  - Process IDs
- Adjustable after container creation



https://docs.docker.com/config/containers/resource_constraints

Container-based Performance Concepts - con’t

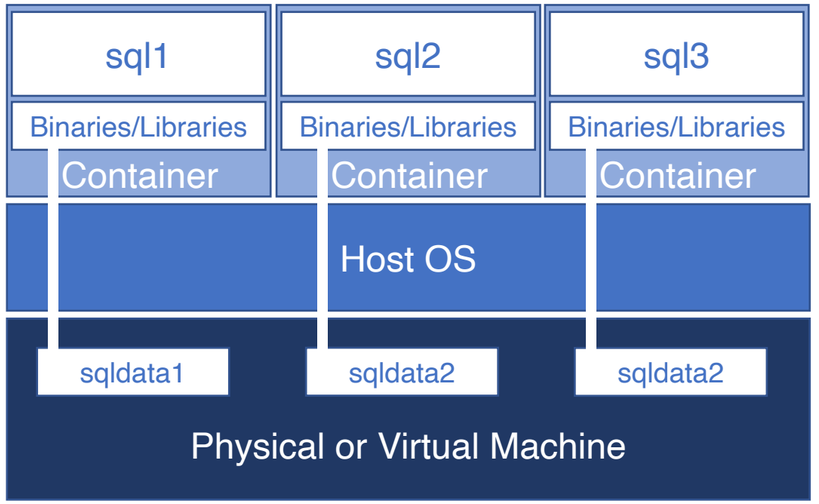
- CPU
  - **CPU Sets** will limit access to specific CPUs
  - **Limits** influence scheduling
  - **Shares** kick in when CPU is constrained
  - SQL Server will see all CPUs
- Memory Limits will limit access
- `mssql-conf` controls SQL Server’s access to memory
- Configuration Best Practices



https://docs.microsoft.com/en-us/sql/linux/sql-server-linux-performance-best-practices

Container-based Monitoring Concepts

- Stabilize the hostname inside the container
  - Enables third party monitoring scenarios
  - DMVs but no WinRM or DCOM/RPC
- docker stats
- Metrics are exposed by docker
- Monitor the base system
- Use restart to keep a container online
  - No, on-failure, always, unless-stopped



Demo!

- **Examine how SQL Server sees the host hardware**
- **Using docker stats to examine performance data**

Review

- **Storing Persistent Data in Containers**
- **Non-root Containers**
- **Custom Container Builds with SQL Server Features and Configuration**
- **Getting Data into Your Containers**
- **Container Performance Concepts**

Need more data or help?

http://www.github.com/nocentino/presentations

Links to resources
Demos
Presentation
Pluralsight

aen@centinosystems.com
@nocentino
www.centinosystems.com

Solving tough business challenges with technical innovation

Centino systems

Thank You!