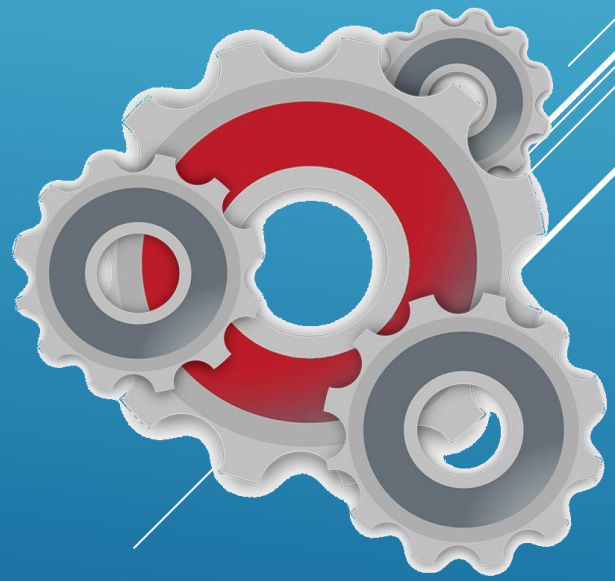
MIGRATING TO CONTAINERS

The Promise And Pitfalls Of Moving To The Cloud

- 20 years programming with relational databases
  - MSSQL, Interbase/Firebird, MySQL, Oracle
- Specialise in manipulating large data sets

NIGEL TAVENDALE

www.allthingssyslog.com



ALL THINGS SYSLOG

CODE SAMPLES AVAILABLE AT:

https://github.com/ntavendale/Ekon24

NIGEL TAVENDALE

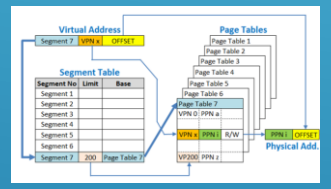
www.allthingssyslog.com

ALL THINGS SYSLOG

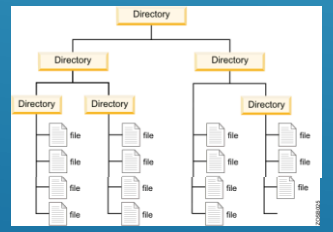
CONTAINERS

Kernel

Memory

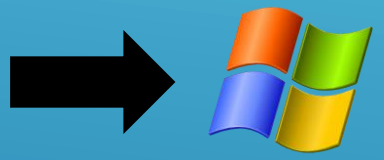


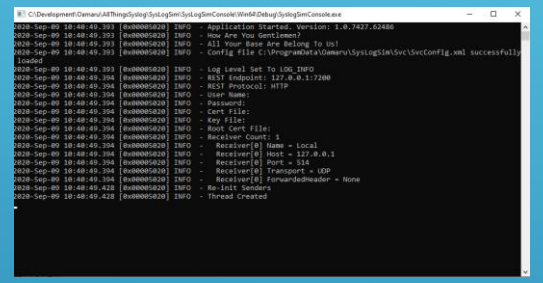
File System



OPERATING SYSTEM

A container creates a virtual Operating System and presents it to your application in a way that the app cannot tell the difference.





OPERATING SYSTEM SUPPORT

To virtualize your OS, and run applications in containers, you need your OS to cooperate.

Linux

- **LXC**. Linux Containers. Open Source. C api.
- **Podman**. No daemon. Runs containers & pods as child processes.
- **Docker**. Originally used LXC. Now uses libcontainer.

Windows

- **Docker**. Windows Server 2016
- **Docker Desktop**. Windows 10

Docker on Windows can run Windows or Linux containers (with WSL). Must use Linux containers to run Kubernetes master node.

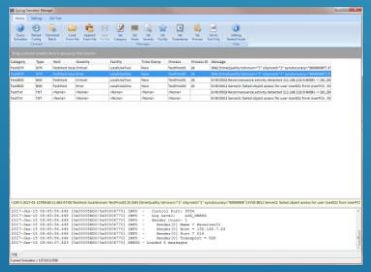
IMAGES & CONTAINERS

A container is a runtime representation of an image file that you define.

```
TStrings IUnknown
    TObjectList
TForm  ILogger
```

Compiler

MyProgram.exe

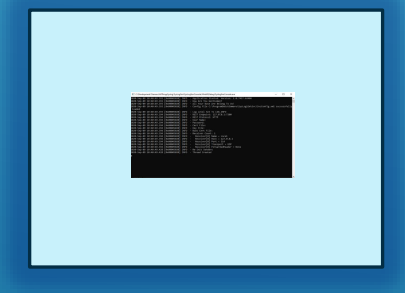


Executables become applications at runtime

```
Base OS Image
Executables
Entry Point
```

Docker Build

myimage:v01



Images become containers at runtime

WINDOWS CONTAINER IMAGES

- No desktop, so no GUI apps.
- Can have services, just not as primary process.
- Windows images don't use latest tag. Have to pull specific build
- Windows Events are local to container. When container goes away, the event log goes with it.

Will Run Delphi Win32 Apps

- Windows: mcr.microsoft.com/windows:2004
- Windows Server Core: mcr.microsoft.com/windows/servercore:ltsc2019

Will Not Run Delphi Win32 Apps

- IOT Core: mcr.microsoft.com/windows/iotcore:1809
- Nanoserver: mcr.microsoft.com/windows/nanoserver:2004

BUILDING IMAGES WITH A DOCKERFILE

FROM Need to tell docker what base image we are starting off with.

RUN Runs a command inside image, if it is available. If .exe not present you must copy it in first.

COPY Copy our exe and files into the container. All paths must be relative to directory in build command. "..\" not allowed

WORKDIR Define working directory for container. Will look here for entry point.

EXPOSE Tell docker which port and protocol application is listening on. Used by other apps inside container.

CMD Define command to run when container starts, and arguments.

```
FROM mcr.microsoft.com/windows/servercore:ltsc2019

RUN mkdir C:\MyApp

COPY Win64/Debug/webapp.exe C:/MyApp

WORKDIR C:/MyApp

EXPOSE 5200/tcp

CMD ["webapp.exe", "0.0.0.0:5200"]
```

BUILD: **`docker build -t delphi_test .`** (Note the "." at the end. Every file must be relative to this)

RUN: **`docker run --name webapp -p 6200:5200 delphi_test`** (If you don't supply name, docker will make one up.)

USING AN ENTRYPOINT

Containers can have a default ENTRYPOINT.

The ENTRYPOINT is the process that runs the executable defined by the CMD in Dockerfile.

In Windows this is `cmd`.

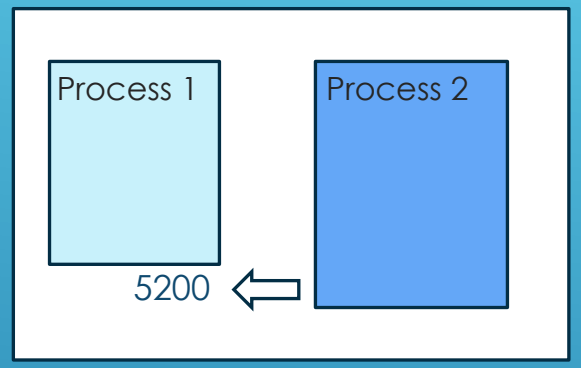
Dockerfile can replace this with our app:

```
ENTRYPOINT ["webapp.exe", "0.0.0.0:5200"]
```

Our app is essentially the shell at this point

NETWORKING

EXPOSE



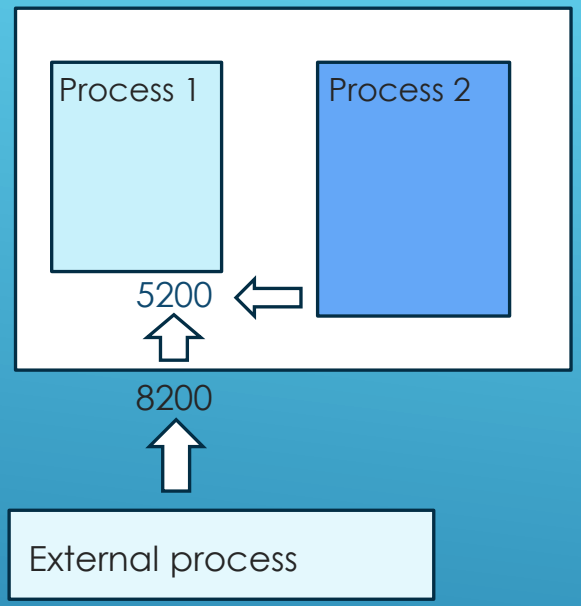
Allows Port/Protocol connections from other processes within the container

Use for inter process communication

Not needed for communicating externally

Default protocol is tcp

Publish



Exposes Port/Protocol to other processes outside the container

Uses –p flag on command line and can remap port

```
-p 8080:5200
```

Port used by external process

Port used by container process

STORAGE

Local Storage Ephemeral file system that is part of container. When the container stops anything written to local storage goes with it.

Volume

- Created independently of containers
- Managed by docker. No windows systems use plugin architecture.
- Can be anything, depends on the plugin. (Docker only has local plugin for windows).
- Directories created in <ProgramData>\Docker\volumes\<volume-name>_data

Bind Mount

- Local folder on host machine only.
- You must specify the path
- UNC file shares must be mapped to drive letters.

LESSONS I LEARNED

Don't use the Windows Event Log. Always log to files, a database, or Syslog. If logging to file make sure filename is unique or make sure only one instance of your container/process can be up at any one time.

Get you build system working to output & deploy the images before you do anything else. You need to form the habit of building and running containers before you check in code.

Use REST for al inter-process communication, even between processes in the same container.

Use a database when data is to be read and written by more than one container process.

MORE INFORMATION

Docker Documentation:

https://docs.docker.com/

ALL THINGS SYSLOG