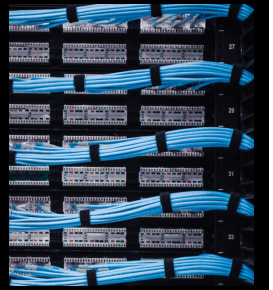
The Value Of Developing With Containers

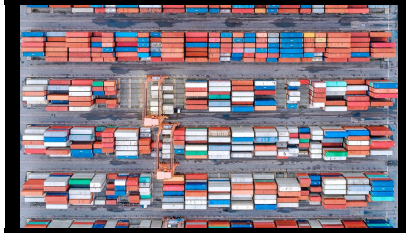
Inspired on Joe Carrol's presentation with the same name

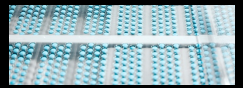
InterSystems®
Health | Business | Government

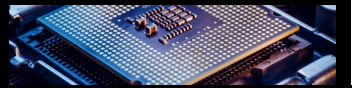
Amir Samary

Manager, Solution Architecture









Shipping in 1812

Shipping in 2019

Standardized

 © InterSystems Corporation. All rights reserved. |

Standardized

 © InterSystems Corporation. All rights reserved. |

Standardized

 © InterSystems Corporation. All rights reserved. |

Container Example

Demo!

```
docker run --init -it --rm --name iris store/intersystems/iris-community:2020.1.0.197.0
```

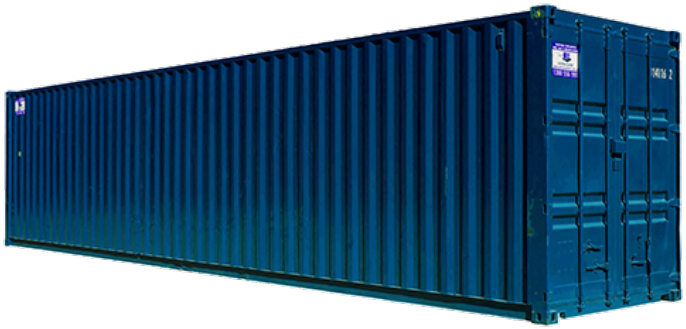
 © InterSystems Corporation. All rights reserved. |

What is a Container

Defining Terms

Image

- Standardized, Portable & Runnable software bundle
  - Application code
  - Dependencies
  - **READ ONLY**



Container

- Image that is Executed in Isolation & Resource Controlled
  - Running Application
  - cgroups
  - linux namespaces

 © InterSystems Corporation. All rights reserved. |

Linux Operating System Terminology

Kernel

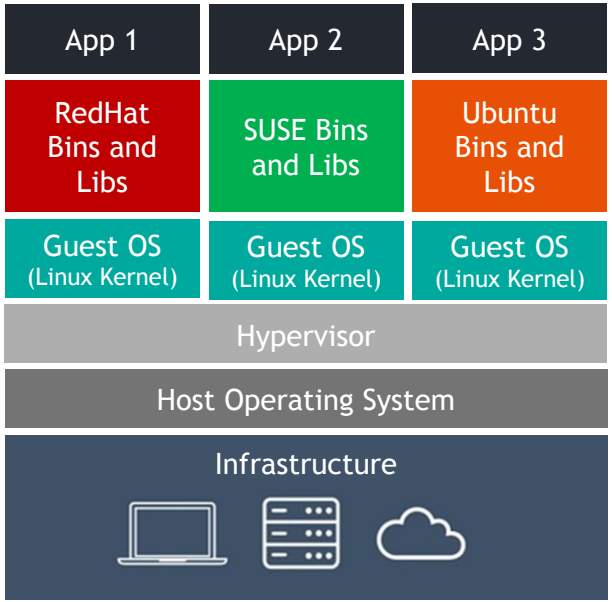
- The kernel is the operating system itself.
- It provides a standard API that allow applications to allocate memory, start processes and threads, access the file system, the network, etc.
- There is only one Linux Kernel out there.
- The kernel runs in a privileged mode because it is the kernel that manages the access to all machine resources

Linux Distribution

- The Kernel alone would be very hard to use.
  - Commands like ls, mkdir, mv, ps, etc. are not part of the kernel.
  - Graphical User Interfaces (GUIs) are not part of the kernel either.
- Anything built on top of this kernel are called “user applications”

 © InterSystems Corporation. All rights reserved. |

Virtual Machines



 © InterSystems Corporation. All rights reserved. |

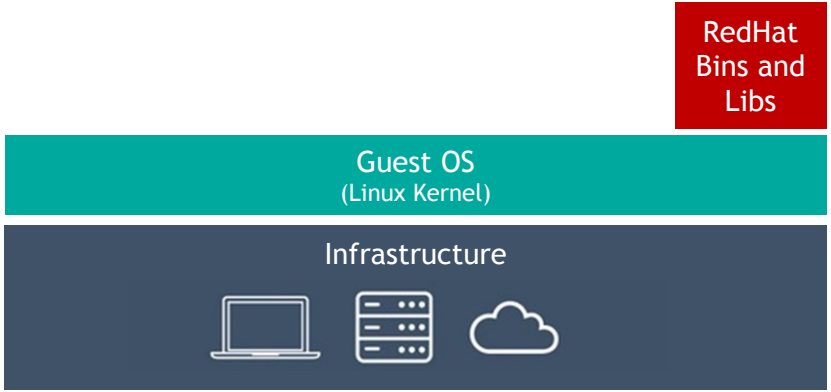
Containers

Image

 © InterSystems Corporation. All rights reserved. |

Containers

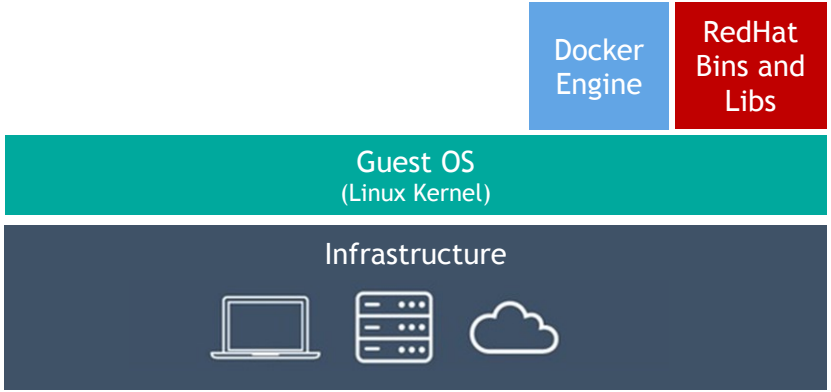
Image



 © InterSystems Corporation. All rights reserved. |

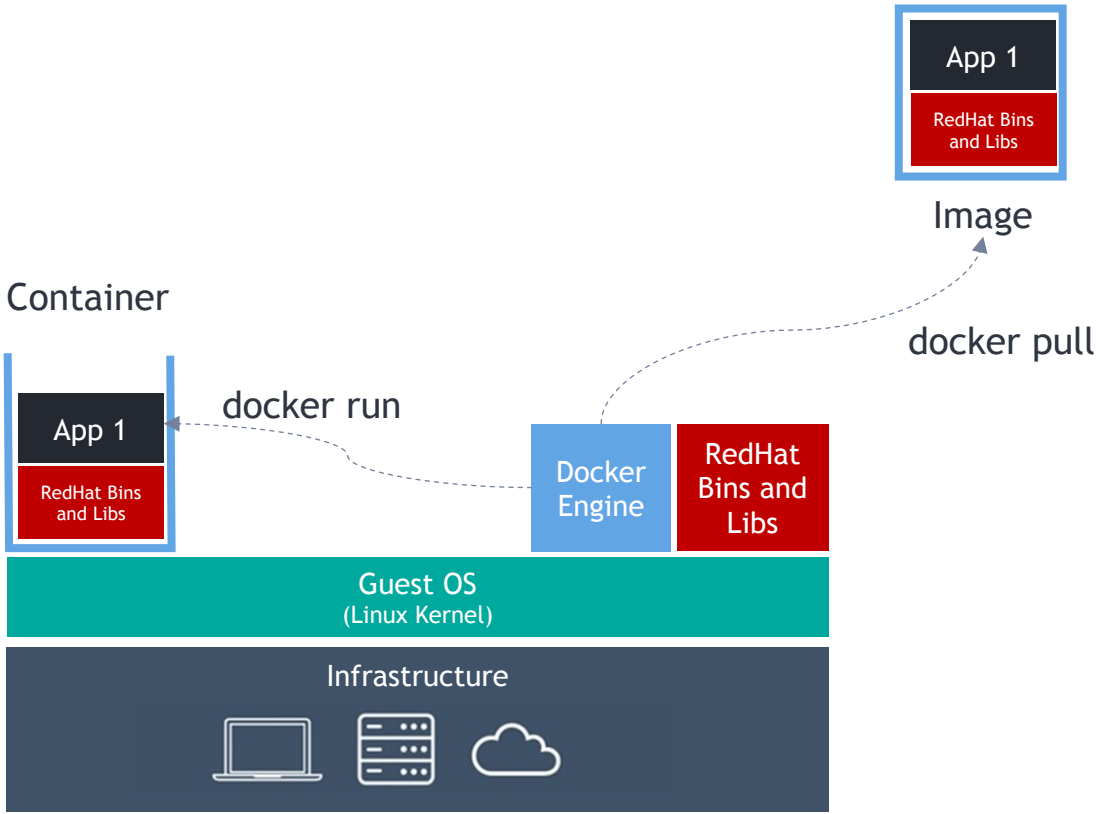
Containers

Image



 © InterSystems Corporation. All rights reserved. |

Containers



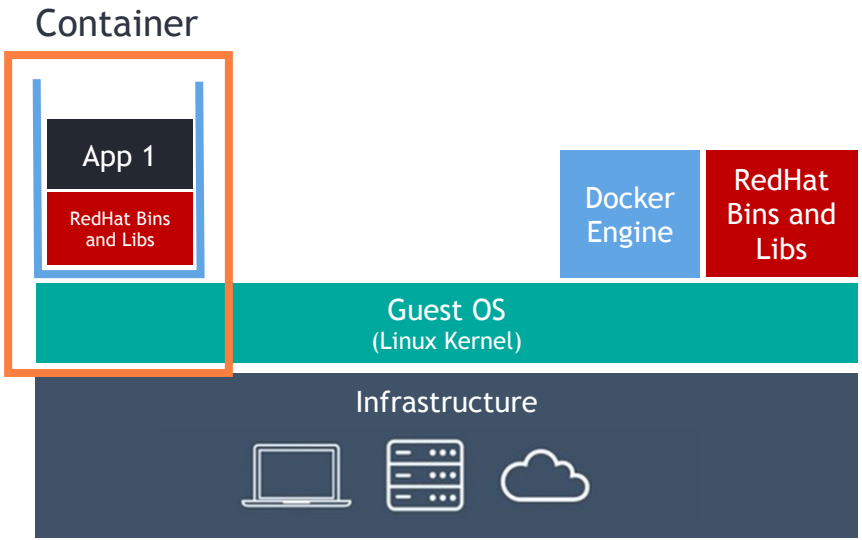
 © InterSystems Corporation. All rights reserved. |

Containers

Image

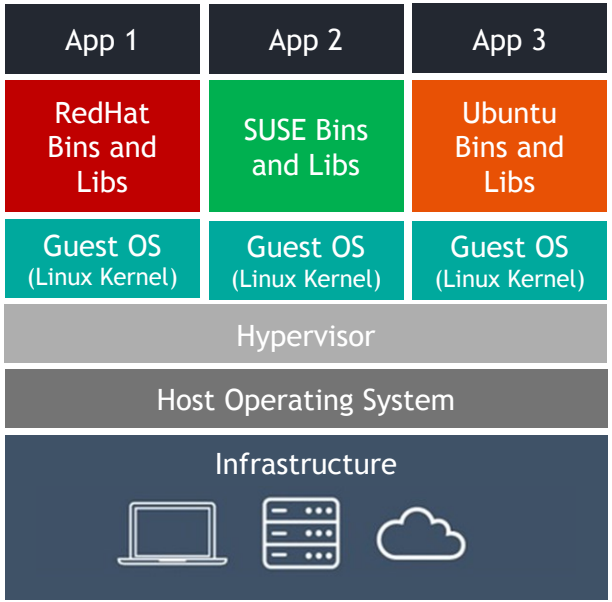
The container uses the same Linux Kernel of the Host!

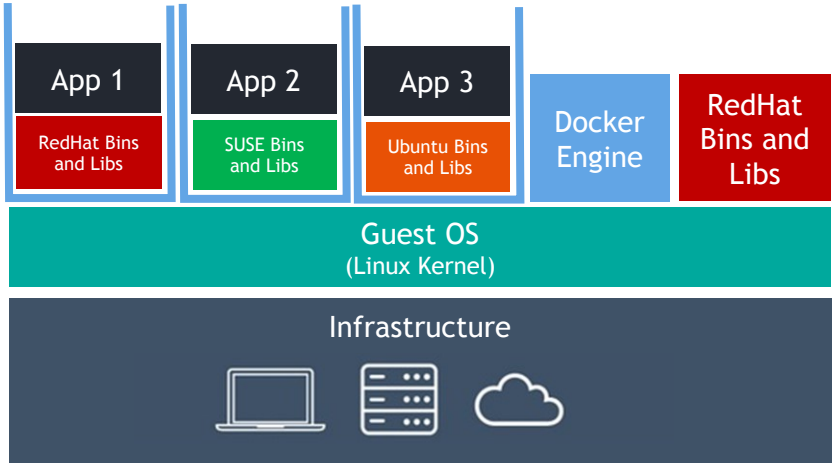
There is no hypervisor! No virtualization!



 © InterSystems Corporation. All rights reserved. |

Virtual Machines x Containers





Containers

 © InterSystems Corporation. All rights reserved. |

Process Isolation Example

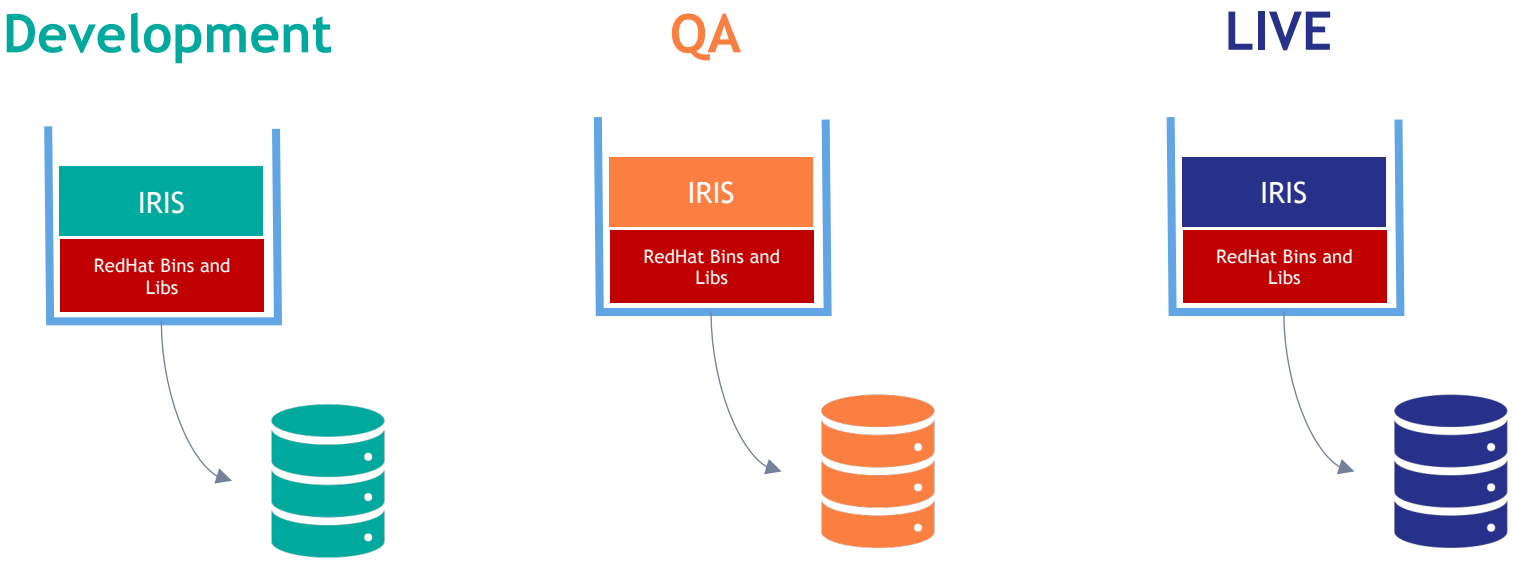
```
docker run --init -it --rm --name iris store/intersystems/iris-community:2020.1.0.197.0
iris list
docker exec –it <container_id> bash
iris list
ps –ef
exit
ps -ef
```

 © InterSystems Corporation. All rights reserved. |

IRIS Containers

Durable %SYS

Container Data is Ephemeral



So, we leave our data on external volumes, outside of the container!

 © InterSystems Corporation. All rights reserved. |

Durable %SYS

New in IRIS

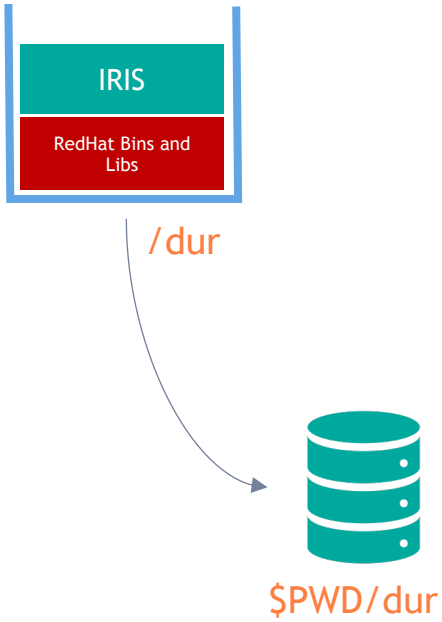
Provides control over where the IRIS configuration is

- Configuration
  - Security Configuration
  - System Database
  - Web Server/Gateway Configuration
  - CPF

Flexibility is necessary!

Durable %SYS Example

```
docker run \
  --volume $PWD/dur:/dur \
  --env ISC_DATA_DIRECTORY=/dur/iris \
  -p 52773:52773 \
  --init -it --rm --name iris \
  store/intersystems/iris-community:2020.1.0.197.0
```

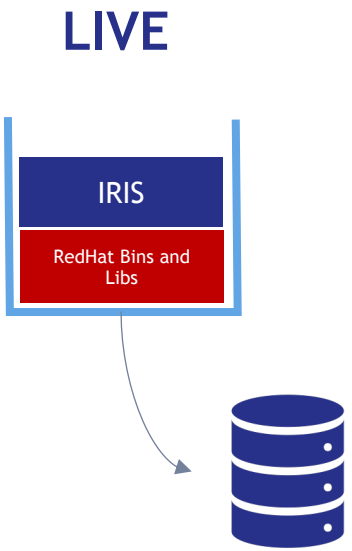


 © InterSystems Corporation. All rights reserved. |

IRIS Containers

Merge CPF

Configuring Instance Parameters



Merge CPF file

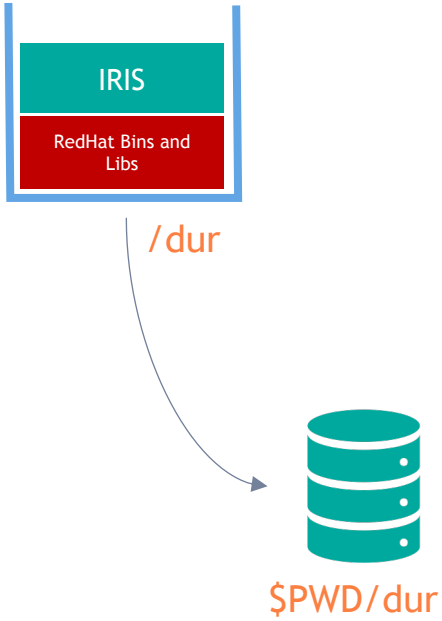
- Defines instance parameters we want
- Each parameter on the merge CPF file replaces the default parameter inside the default CPF file.
- It only works if you use Durable %SYS

```
[config]
globals=0,0,1024,0,0,0
routines=512
```

 © InterSystems Corporation. All rights reserved. |

Durable %SYS Demo

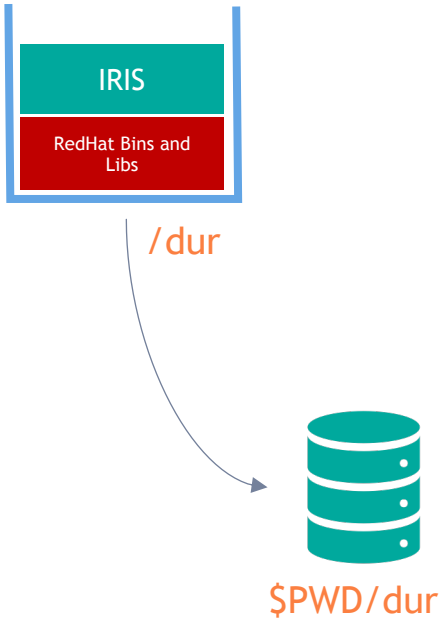
```
docker run \
  --volume $PWD/dur:/dur \
  --env ISC_DATA_DIRECTORY=/dur/iris \
  -p 52773:52773 \
  --init -it --rm --name iris \
  store/intersystems/iris-community:2020.1.0.197.0
```



 © InterSystems Corporation. All rights reserved. |

Merge CPF Example

```
docker run \
  --volume $PWD/dur:/dur \
  --env ISC_DATA_DIRECTORY=/dur/iris \
  --env ISC_CPF_MERGE_FILE=/dur/merge.cpf
  -p 52773:52773 \
  --init -it --rm --name iris \
  store/intersystems/iris-community:2020.1.0.197.0
```



 © InterSystems Corporation. All rights reserved.

IRIS Containers

iris-main

Still More Flexibility Needed

- Is IRIS up or down?
- License Key?
- Script to run before/after IRIS starts?

 © InterSystems Corporation. All rights reserved. |

iris-main

iris-main routine

- Orchestrates IRIS in the container

When an Image becomes a Container

iris-main

- Provides **run time** flexibility
  - IRIS up or down
  - Run scripts before/after start up
  - Pass in a License key
- By default
  - Starts IRIS and waits for a signal to stop IRIS

iris-main example: Defining the Instance's Password

```
docker run \
    --volume ~/demo/dur:/dur \
    --env ISC_DATA_DIRECTORY=/dur/iris \
    store/intersystems/iris-community:2019.3.0.302.0 \
    -p /dur/password.txt
```

 © InterSystems Corporation. All rights reserved. |

iris-main example: Assigning a License Key

```
docker run \
  --volume $PWD/dur:/dur \
  --env ISC_DATA_DIRECTORY=/dur/iris \
  --env ISC_CPF_MERGE_FILE=/dur/merge.cpf \
  -p 52773:52773 \
  --init -it --rm --name iris \
  amirsamary/irisdemo:iris.2020.1.0.197.0 \
  -p /dur/password.txt \
  -k /dur/iris.key
```

 © InterSystems Corporation. All rights reserved. |

Recap

Takeaways

Image

- Union of **Code** and **Dependencies**
- Standardized, Portable & Runnable software bundle
  - **READ ONLY**

Container

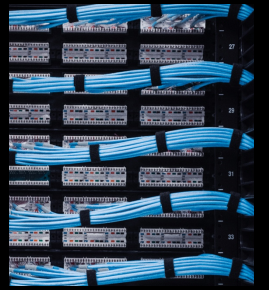
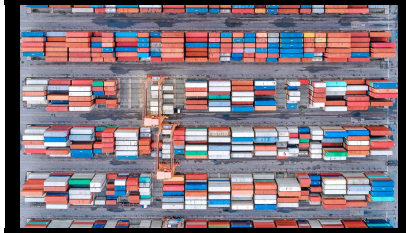
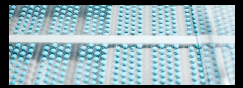
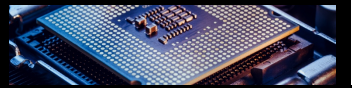
- Image that is Executed in Isolation and Resource Controlled

Durable %SYS and iris-main

- You make decisions about how IRIS runs when you start a container

https://bit.ly/38ifbg6

The power behind what matters.

InterSystems®
Health | Business | Government

Thank you.