Science and Technology Facilities Council

ISIS Neutron and Muon Source

isis.stfc.ac.uk

Containerised Control Systems Development at Isis and Potential Use in an Epics System

WEPV050

gareth.howells@stfc.ac.uk

G.D. Howells, I.D. Finch, ISIS Neutron and Muon Source, Didcot, United Kingdom

ICALEPCS 2021

ISIS Overview

ISIS has a long history in the neutron arena, supporting a scientific user program. Control of the accelerator is currently via Vsystem control software on Itanium hardware, which is coming to the end of it's supported life. ISIS is therefore looking to update the control system and join the open source EPICS community. We outline a brief summary of how ISIS will embark on this transition project which will lead to Docker being used as a core component. Normal operations must continue while we progress towards the ultimate goal of a full EPICS control system.

Application of Docker Technology at ISIS.

- Docker utilized as a development tool to extract data from Vsystem using MQTT protocol (containers running Mosquitto, CouchDB, Telegraf, InfluxDB and Grafana (TIG stack)).
- Syslog project developed with TIG stack. Telegraf aggregator configured to receive additional data from accelerator and record generic system metrics (including Docker itself).
- Multiple projects established with Docker and Visual Studio Code at the heart of the development (Syslog, PVEcho, Vsystem, MQTT, EPICS, RONA).
- Docker images of EPICS base and support modules compiled on numerous OS (Centos, Debian, Tiny Core, Alpine and Rocky).
- C and Python EPICS projects developed by the group, run as Docker containers, mirroring Vsystem data as EPICS PVs (PVEcho). GitLab used for CI/CD pipelines and Docker image registry.
- Initial conversion to hybrid control system with Vsystem, EPICS and Docker technologies working in harmony, Phoebus and Vsystem control screens for users. Long term migration of Vsystem components to native EPICS, resulting in the end goal of a full EPICS control system with Phoebus GUI.

Data Logging

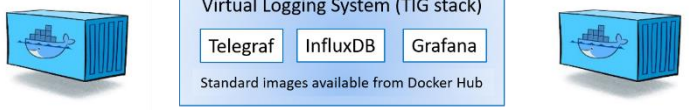
Monitoring Health of Computer Systems

Requirement : Capture system metrics and System Logging Protocol style messages (syslog) from computers and control systems to aid fault diagnosis.

This virtual logging project was used as a route to learning Docker principles, using standard images pulled from Docker Hub. Being ideally suited to run software stacks, Docker was used to investigate aggregator, database and visualisation packages such as Telegraf, FluentD, Fluent-Bit, NodeRed, Rsyslog, InfluxDB, Elasticsearch, Chronograph and Grafana. Previous work within the group had already established Telegraf and InfluxDB as a versatile pairing to capture MQTT topics/data from the ISIS Vsystem. It was therefore a natural step to extend the Telegraf configuration to capture syslog messages, metrics and additional ISIS information, storing the data in InfluxDB. Web based dashboards were composed with Grafana to visualise the data captured.

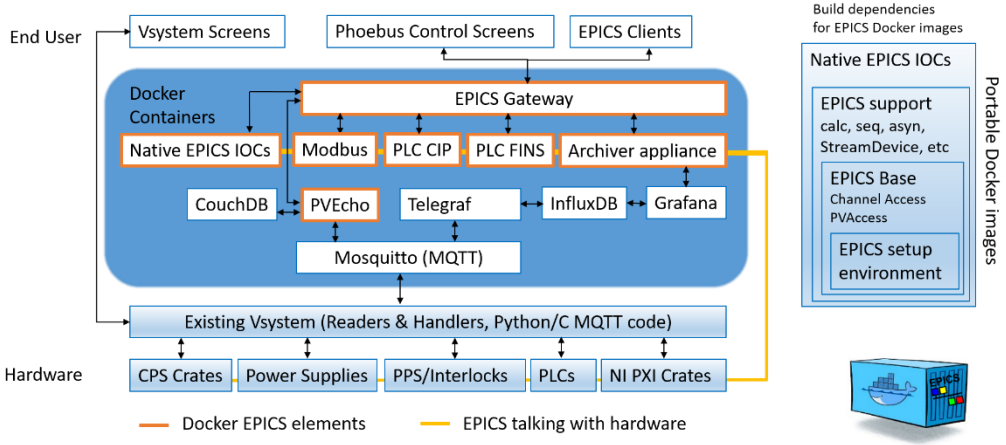
Notable points of interest.

- InfluxDB is available as two distinct versions, both available as Docker images.
- Grafana can query data from InfluxDB using InfluxQL and Flux.
- LDAP (Lightweight Directory Access Protocol) authentication and authorization used for access control to Grafana.
- Recent developments created a customized Grafana Docker image to include a data source for the EPICS Archive Appliance.
- InfluxDB version 2 and Telegraf have provided a method to run Telegraf configurations from a central store.
- InfluxDB databases (also known as buckets) can be configured to have a retention policy, thereby policing database size.



EPICS Docker Model

Model of Proposed Hybrid ISIS Control System with Docker Containers, EPICS and Vsystem



Conclusions

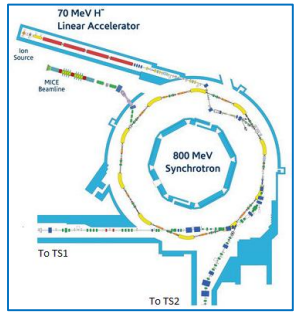
Advantages		Caveats
Rich set of Docker images available from Docker Hub. Container assembly with docker-compose creates very capable software stacks. Visual Studio Code is an elegant integrated development environment under Windows.		Subtle differences exist between the OS on Docker containers and full/virtual Linux systems that need to be appreciated (no kernel, services, PID 1). Windows and Linux Docker engines are also not equivalent (swarm single node mode, network, WSL2 limitations).
Numerous ways to run/configure containers via Docker CLI, Docker dashboard and Portainer (web based GUI for Docker container control).		Large Docker images are an issue, but can be mitigated with multi stage Docker build files.
Custom compiled images can be created with bespoke Docker build files (with the aid of tools such as git, wget, curl).		Docker engine is a single point of failure for running containers. Reliability to date is positive, CPU load being monitored.
Images available easily across development team via standard git repository means or as pre-compiled units e.g. Docker Hub, GitLab registry.		Compiling EPICS support modules can be challenging with Alpine Linux. Considering Debian as the native OS for built images (as tried by other facilities/developers).
Docker swarm mode offers redundancy across multiple Docker engine hosts. ISIS is looking to deploy containers across a Linux cluster with Docker configured in this way.		Running containers that have a desktop style GUI component can be achieved via X-Windows, but is not the most harmonious arrangement and there is processing overhead. Web based containers appear to run efficiently (e.g. Grafana, InfluxDB).
EPICS components can be successfully built as Docker images. Careful selection of which EPICS components to include lead to a set of complimentary Docker images for development and production (core setup, base, support, FINS, CIP, Modbus). Facilities can define their own EPICS Docker image requirements.		

Science and Technology Facilities Council

ISIS Neutron and Muon Source

Docker at ISIS

isis.stfc.ac.uk



ISIS has a long history in the neutron arena, supporting a scientific user program. Control of the accelerator is currently via Vsystem control software on Itanium hardware, which is coming to the end of it's supported life. ISIS is therefore looking to update the control system and join the open source EPICS community. We outline a brief summary of how ISIS will embark on this transition project which will lead to Docker being used as a core component. Normal operations must continue while we progress towards the ultimate goal of a full EPICS control system.

Application of Docker Technology at ISIS.

- Docker utilized as a development tool to extract data from Vsystem using MQTT protocol (containers running Mosquitto, CouchDB, Telegraf, InfluxDB and Grafana (TIG stack)).
- Syslog project developed with TIG stack. Telegraf aggregator configured to receive additional data from accelerator and record generic system metrics (including Docker itself).
- Multiple projects established with Docker and Visual Studio Code at the heart of the development (Syslog, PVEcho, Vsystem, MQTT, EPICS, RONA).
- Docker images of EPICS base and support modules compiled on numerous OS (Centos, Debian, Tiny Core, Alpine and Rocky).
- C and Python EPICS projects developed by the group, run as Docker containers, mirroring Vsystem data as EPICS PVs (PVEcho). GitLab used for CI/CD pipelines and Docker image registry.
- Initial conversion to hybrid control system with Vsystem, EPICS and Docker technologies working in harmony, Phoebus and Vsystem control screens for users. Long term migration of Vsystem components to native EPICS, resulting in the end goal of a full EPICS control system with Phoebus GUI.

Science and Technology Facilities Council
ISIS Neutron and Muon Source

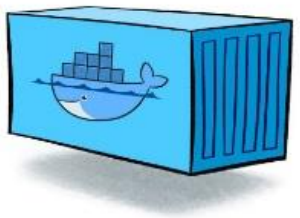
Docker : Logging

Monitoring Health of Computer Systems

Requirement : Capture system metrics and System Logging Protocol style messages (syslog) from computers and control systems to aid fault diagnosis.

This virtual logging project was used as a route to learning Docker principles, using standard images pulled from Docker Hub. Being ideally suited to run software stacks, Docker was used to investigate aggregator, database and visualisation packages such as Telegraf, FluentD, Fluent-Bit, NodeRed, Rsyslog, InfluxDB, Elasticsearch, Chronograph and Grafana. Previous work within the group had already established Telegraf and InfluxDB as a versatile pairing to capture MQTT topics/data from the ISIS Vsystem. It was therefore a natural step to extend the Telegraf configuration to capture syslog messages, metrics and additional ISIS information, storing the data in InfluxDB. Web based dashboards were composed with Grafana to visualise the data captured.
Notable points of interest.

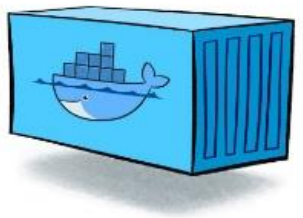
- InfluxDB is available as two distinct versions, both available as Docker images.
- Grafana can query data from InfluxDB using InfluxQL and Flux.
- LDAP (Lightweight Directory Access Protocol) authentication and authorization used for access control to Grafana.
- Recent developments created a customized Grafana Docker image to include a data source for the EPICS Archive Appliance.
- InfluxDB version 2 and Telegraf have provided a method to run Telegraf configurations from a central store.
- InfluxDB databases (also known as buckets) can be configured to have a retention policy, thereby policing database size.



Virtual Logging System (TIG stack)

Telegraf | InfluxDB | Grafana

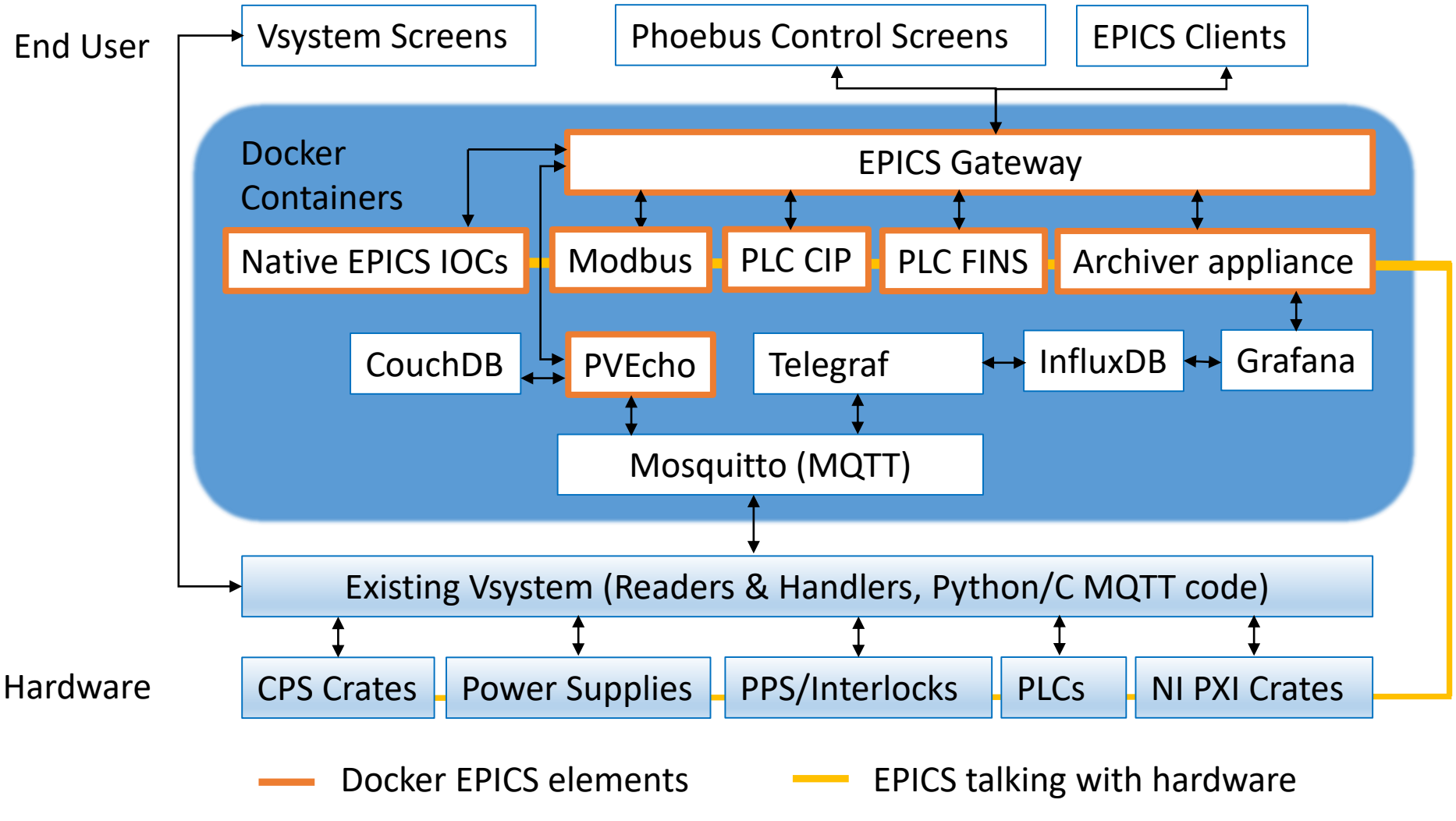
Standard images available from Docker Hub



Science and Technology Facilities Council
ISIS Neutron and Muon Source

Docker : EPICS

Model of Proposed Hybrid ISIS Control System with Docker Containers, EPICS and Vsystem



Build dependencies for EPICS Docker images

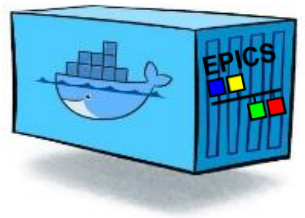
Native EPICS IOCs

EPICS support
calc, seq, asyn, StreamDevice, etc

EPICS Base
Channel Access
PVAccess

EPICS setup environment

Portable Docker images



Science and Technology Facilities Council
ISIS Neutron and Muon Source

Docker : Conclusions

Advantages		Caveats
Rich set of Docker images available from Docker Hub. Container assembly with docker-compose creates very capable software stacks. Visual Studio Code is an elegant integrated development environment under Windows.		Subtle differences exist between the OS on Docker containers and full/virtual Linux systems that need to be appreciated (no kernel, services, PID 1). Windows and Linux Docker engines are also not equivalent (swarm single node mode, network, WSL2 limitations).
Numerous ways to run/configure containers via Docker CLI, Docker dashboard and Portainer (web based GUI for Docker container control).		Large Docker images are an issue, but can be mitigated with multi stage Docker build files.
Custom compiled images can be created with bespoke Docker build files (with the aid of tools such as git, wget, curl).		Docker engine is a single point of failure for running containers. Reliability to date is positive, CPU load being monitored.
Images available easily across development team via standard git repository means or as pre-compiled units e.g. Docker Hub, GitLab registry.		Compiling EPICS support modules can be challenging with Alpine Linux. Considering Debian as the native OS for built images (as tried by other facilities/developers).
Docker swarm mode offers redundancy across multiple Docker engine hosts. ISIS is looking to deploy containers across a Linux cluster with Docker configured in this way.		Running containers that have a desktop style GUI component can be achieved via X-Windows, but is not the most harmonious arrangement and there is processing overhead. Web based containers appear to run efficiently (e.g. Grafana, InfluxDB).
EPICS components can be successfully built as Docker images. Careful selection of which EPICS components to include lead to a set of complimentary Docker images for development and production (core setup, base, support, FINS, CIP, Modbus). Facilities can define their own EPICS Docker image requirements.		