



# EPICS Deployment at Fermilab

Pierrick Hanlet

21 September 2022

A Partnership of:

US/DOE

India/DAE

Italy/INFN

UK/UKRI-STFC

France/CEA, CNRS/IN2P3

Poland/WUST



# Fermilab Control System Content

- Introduction
- Motivation
- Continuous Integration/Continuous Deployment (CI/CD)
- Testing
- Template IOCs
- IOC Deployment and Monitoring
- Security
- Present & Future Work
- Summary



# Introduction

- Historically, Fermilab is not an EPICS house
  - Previous efforts to deploy EPICS
  - Some experiments have partial controls using EPICS
  - 3 years ago, there was a lot of eye-rolling when I mentioned EPICS
- Fermilab is building a new proton source and superconducting linear accelerator
  - PIP-II
    - Early on, the leadership stated that PIP-II's control system would be EPICS
    - Also upgrading the controls infrastructure for the remainder of the complex
    - New system will be hybrid (at least for the foreseeable future)
- Fermilab controls network has multiple VLANs – use multicast
- Little EPICS expertise, so want to make IOC development easy

# Motivation

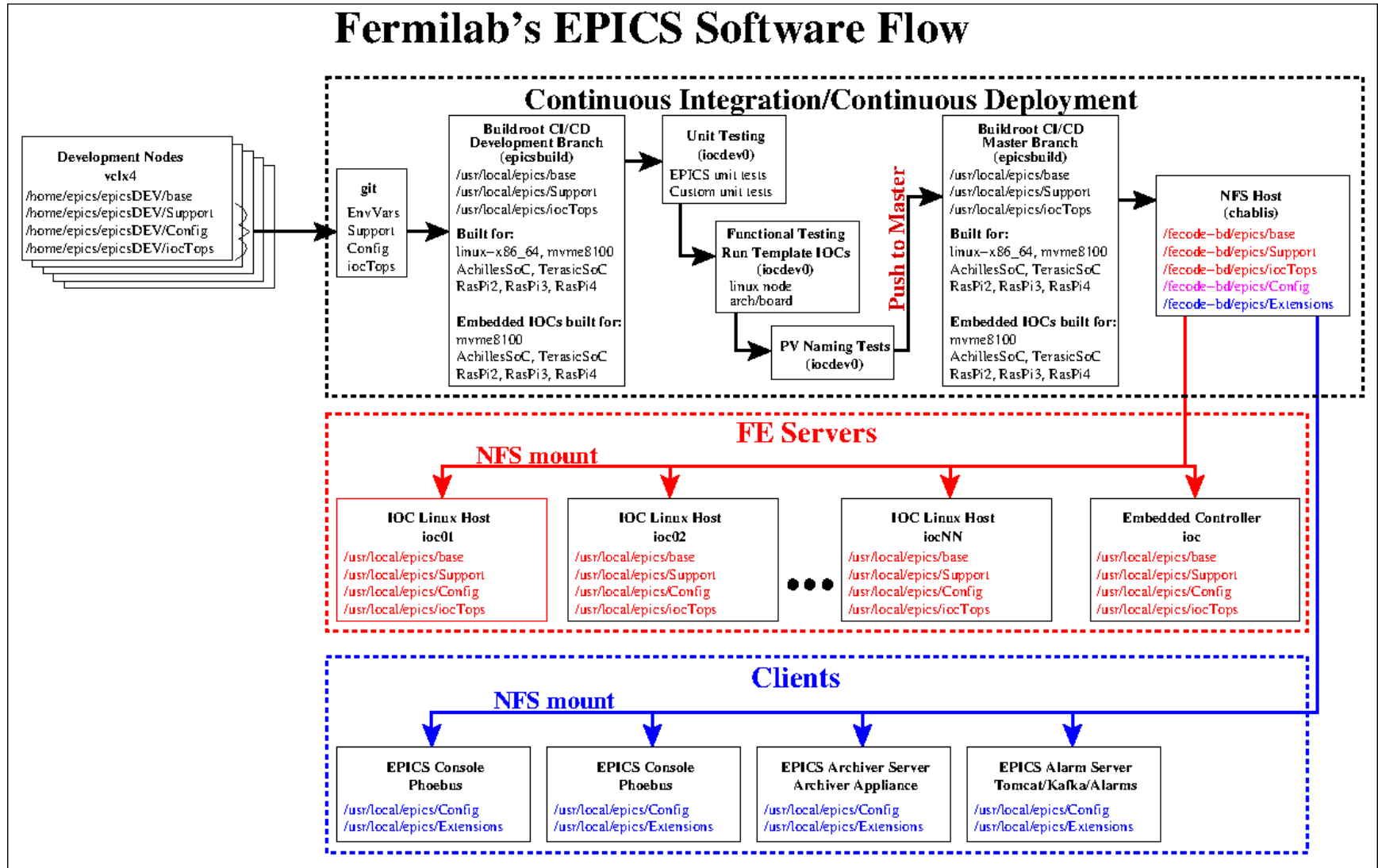
- Treating EPICS deployment as a green field to simplify deployment for non-experts
  - <https://ghe-pip2.fnal.gov/epics-controls/>
- Small controls team, therefore we:
  - require robust build of infrastructure
  - require automated build procedures
  - require extensive testing
  - require minimal functionality to automate deployment/monitoring of IOCs
- Developed a standard EPICS infrastructure to simplify developing IOCs for new developers
  - “base” and “Support” software are built (on all supported platforms) and made available on controls network
  - developers start from template IOCs
  - template IOCs have minimal basic functionality required of all FNAL IOCs
- Automated build for ease and robustness
- Standard deployment for ease in maintaining software
- Standard deployment to simplify debugging
- Options – e.g. containers – kept open for supporting legacy hardware
- Implement modern computing practices Continuous Integration/Continuous Deployment (CI/CD)
- Use PVXS for instrumentation and high rate/high volume data and CA for slow controls

# FNAL EPICS code structure

- Structure follows conventional EPICS implementations
- The Fermilab “standard deployment” of EPICS code assumes a 3-tier build:
  - EPICS base – Production code resides in ***/usr/local/epics/base***
  - EPICS Support – Production code resides in ***/usr/local/epics/Support***
  - EPICS IOCs – Production code resides in ***/usr/local/epics/iocTops***
- EPICS base, Support, & iocTops are built for different architectures/platforms
  - Presently have builds for: linux-x86\_64, arm/Cyclone-V, arm/Arria-10, arm/RasPi2, arm/RasPi3 arm/RasPi4, ppc/mvme8100
- Code base is built and tested in Continuous Integration/Continuous Deployment processes
- EPICS base, Support, & iocTops are hosted by NFS server
- Goal is to have a robust EPICS code base to simplify novice EPICS-developer deployment and to help experts in debugging



# CI/CD – Software Path to Deployment



# CI/CD – Deployment of New IOCs

- Implementation developed by Mariana González.
- Using Github for code management, documentation and issue tracking.
- Using Github Actions tool to automate building and testing
- Using Buildroot for configuring and cross-compiling
- Full CI/CD chain is complete and successfully tested
  - Code and documentation migrated Github
  - Automated build of the 3-tier EPICS in all supported architectures
  - Automated unit testing – working for host architecture
  - Automated testing:
    - Unit tests
    - Basic functional tests (based on templateIOC tests)
    - Check for duplicate PV names
    - Passing tests allows for code to be build, tagged, and pushed to NFS host
  - IOC owner must register IOC before deploying



# CI/CD – Pipeline

Triggered via push 23 days ago

Status

Total duration

Artifacts

hanlet pushed → 26ed3a7 master

Success

4m 5s

1

## build.yml

on: push



### build

succeeded 23 days ago in 1m 17s

```
68 make -C ./configure install
69 make -C 0.linux-x86_64 -f ../Makefile TOP=../.. \
70   T_A=linux-x86_64 install
71 make[2]: Entering directory `/scratch/products/epi
72 perl -CSD /usr/local/epics/base/bin/linux-x86_64/m
73 make[2]: Entering directory `/scratch/products/epi
74 perl -CSD /usr/local/epics/base/bin/linux-x86_64/m
75 perl -CSD /usr/local/epics/base/bin/linux-x86_64/m
```

### Artifacts

Produced during runtime

Name

 iocTops-R7.0.5-173

25.3 MB

### test

succeeded 23 days ago in 1m 49s

```
42
43 Scenario Outline: IOCS heartbeats increase over time. # fe
44   Given <ioc> IOC # fe
45   When I make a request for the <pv> heartbeat # fe
46   And I wait 15 seconds # fe
47   And I make a request for a second <pv> heartbeat value # fe
48   Then I receive a value incremented by 1 # fe
49
50 Examples:
51   | ioc | pv |
52   | templateRaspi4 | FNAL:IOC_TEST_RASPI4_01:ALIVE |
53   | templateAchilles | FNAL:IOC_TEST_ACH_01:ALIVE |
54   | templatePIRM4 | FNAL:IOC_TEST_PIRM4_01:ALIVE |
55   | templateRasPi2 | FNAL:IOC_TEST_RASPI2_01:ALIVE |
56   | templateRasPi3 | FNAL:IOC_TEST_RASPI3_01:ALIVE |
57 Channel connect timed out: 'FNAL:IOC_TEST_SOC:ALIVE' not found.
58 Channel connect timed out: 'FNAL:IOC_TEST_SOC:ALIVE' not found.
59   | templateSoC | FNAL:IOC_TEST_SOC:ALIVE |
60
```



# CI/CD – Pipeline



Instant feedback

## Trace-ability

hanlet added 3 commits 21 days ago

- o Add HB650acnet 491d0a0
- o Added HB650acnet to Makefile X 74d24e4
- o While developing, I was pointing to a development version of CMcryo; ... ✓ 666509a

mariana merged commit 44cc39b into master 21 days ago [Hide details](#) [Revert](#)

3 checks passed		
✓	build build	<a href="#">Details</a>
✓	test test	<a href="#">Details</a>
🕒	deploy	<a href="#">Details</a>



# Template IOCs

Each Fermilab IOC running on the controls network will provide the following:

- heartbeat
- IOC statistics (CPU usage, memory usage, etc.)
- capability of 20 Hz scan rate
- use aSub (specific Support module) record for interfacing with IOC specific custom code (C/C++ libraries)
- access to acnet (acnetPV wrapper)
- reccaster

I propose to use the template in the following steps:

- build and use the IOC as is to test connectivity to your embedded system
- once established, create IOC specific PVs, code, and link to IOC specific custom libraries

# Scope of Testing

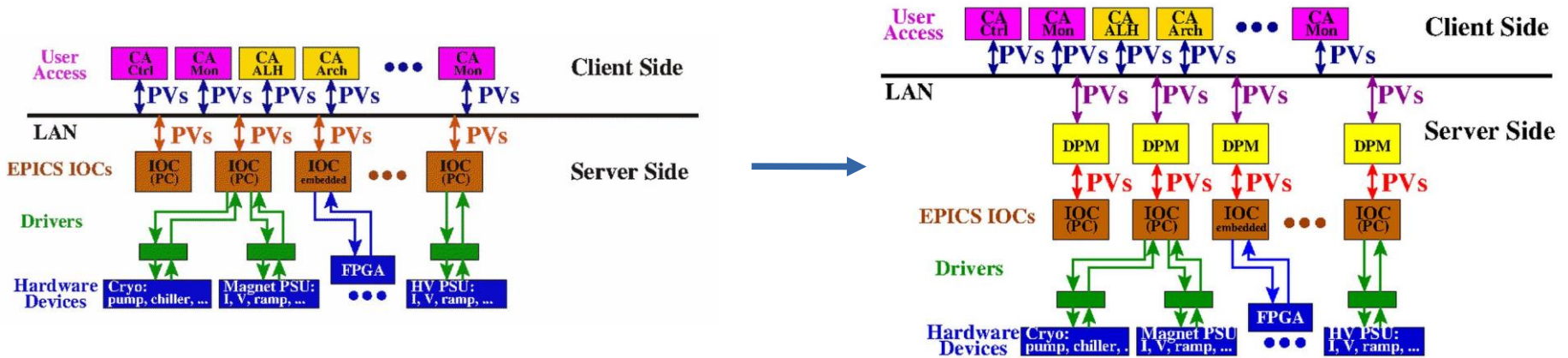
In order to deliver a robust framework for the controls system, the tests we intend to implement are:

- Unit tests
- Functional tests
- Stress tests:
  - Check connection loss of NFS host
  - Scan rate
  - Data transfer bandwidth
- Operational:
  - Register IOCs when coming online
  - Monitor IOC for monitoring registered IOCs

# Security

By default, EPICS IOCs, once deployed on the network, make all their PVs accessible to any other EPICS process. We want to limit this capability by restricting access using currently implemented kerberos authentication. To accomplish this:

- Use existing EPICS capability – “access security”
- Restrict IOCs to single restricted account name
- Restrict access to IOCs via DPMs
- Access to DPMs require kerberos authentication from client side



# Documentation and Versioning

- Presently documented on Github
- Source code versioning in Github
- Git Branch model:
  - Using a simple branch model
  - We consider origin/master to be the branch where the source code always reflects a production-ready state.
  - Using Github capabilities of branch protection for master
  - Other branches should contain the work of new code, new features and bug fixes and will be merged back into the master branch when the work is finished and properly reviewed & tested.
- Code/configurations and build scripts in Github Actions
- Will keep 3 executable versions, change is through softlinks

# Work in Progress

- improve build time (base to iocTops takes ~4 hrs) on a VM
  - moving buildroot to a beefier machine – “epicsbuild” – in progress
- more functional test content
- add security scans
- develop tests (post-build - pre-deployment) - I imagine this to continue for years
- establish network boot for the raspberrypis and TerasicSoC - complete for the AchillesSoC
- more documentation
- continue work with SoC/SoM architectures/platforms
  - FPGA data to redis, IOC on external server
  - FPGA data to “/dev/mem”, IOC on arm processor
  - Presently have EPICS builds for: TerasicSoc (Cyclone V) and Achilles (Arria-10)
- Will add Xilinx based SoMs: proposing MiniZed (Zynq 7Z007S), ZedBoard (Zynq-7000), TE0803 (Zynq UltraScale)



# Summary

- Fermilab is finally an EPICS house
- Green Field – want robust, reproducible, versioned, and easily maintained
- Template IOCs developed to:
  - help new EPICS developers
  - ensure required basic functionality of all IOCs
- Software is built for several architectures/boards
- CI/CD path is in place and operational
- More tests need developing

