



EPICS Deployment at Fermilab

Pierrick Hanlet

26 April 2023

A Partnership of:
US/DOE
India/DAE
Italy/INFN
UK/UKRI-STFC
France/CEA, CNRS/IN2P3
Poland/WUST



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)
- Using PVAccess for higher rate data

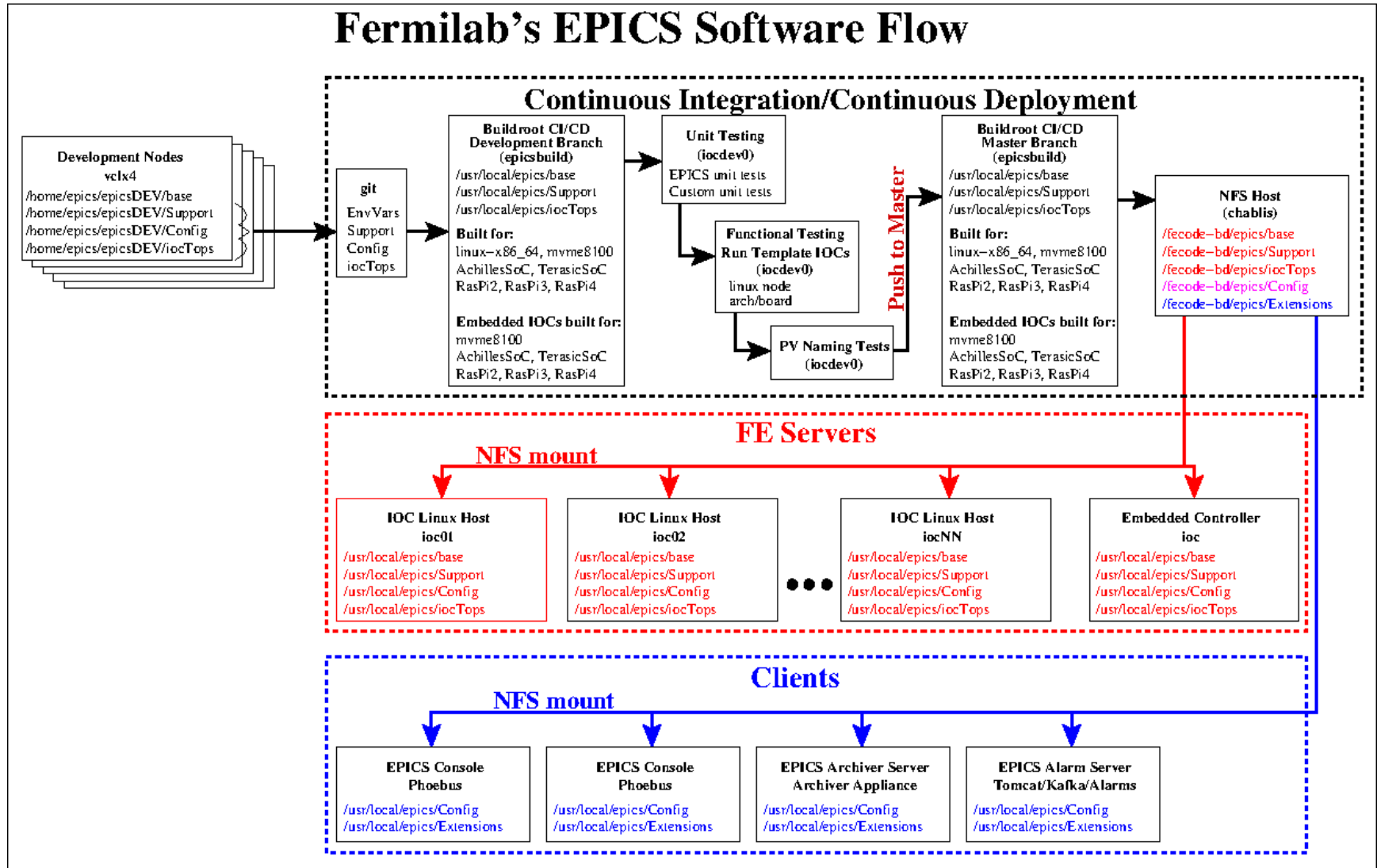
FNAL EPICS code structure

- Structure follows conventional EPICS implementations
- The Fermilab “standard deployment” of EPICS code assumes a 3-tier IOC 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

FNAL EPICS code structure

- The Fermilab “standard deployment” of EPICS configuration code:
 - EPICS base – Production code resides in ***/usr/local/epics/Config***
- 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



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

FNAL EPICS code structure

- The Fermilab “standard deployment” of EPICS configuration files:
 - EPICS base – Production code resides in */usr/local/epics/Config*
 - Alarms
 - Archiving
 - ...

FNAL EPICS Infrastructure

- Registering IOCs
- Configure EPICS tools to use multicast instead of broadcast
- Xpra - accelerating pushing X displays or other alternatives
- DPM - can show both ACNET and EPICS data
- using Kubernetes for ChannelFinder, Elasticsearch (for ChannelFinder and alarms), and Kibana (also for alarms)
- Combining EPICS and ACNET alarms server side (client side already works)
- Pheobus plugins for ACNET