Debian Packaging at NSLS-II

Anton Derbenev

Ian Roi Talla

Oksana Ivashkevych

EPICS Collaboration Meeting - 2019

BROOKHAVEN
NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY

BROOKHAVEN SCIENCE ASSOCIATES

Repository Status

- NSLS-II public Debian repository mirror is available
  - https://epicsdeb.bnl.gov/debian/
- Updated to include recent contributions to the **epicsdeb** project on GitHub
  - https://github.com/epicsdeb
- Notable news:
  - Debian 9 (stretch) packaging is now available
  - For stretch and further, 32-bit builds were dropped and packages are no longer provided
  - For stretch and further, the RTEMS build profile is not used and rtems-specific packages are no longer provided

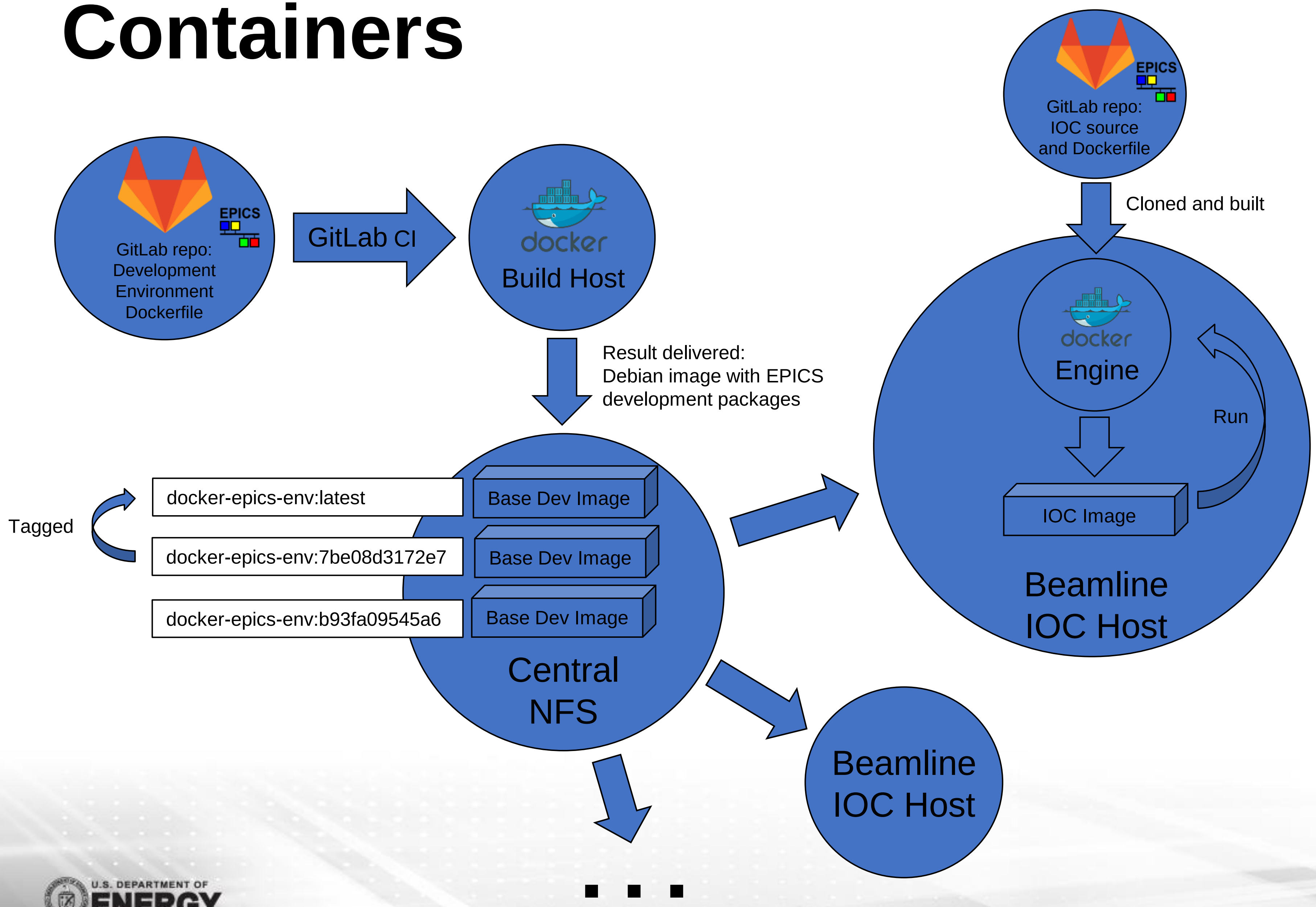
Support Plans

- We will continue our efforts to accommodate and contribute to GitHub updates and to keep the mirror up
- As always, thanks to all **epicsdeb** contributors for bringing new upstream versions, patches, and pull requests
- There are still outstanding pull requests – code was built and packages are available in staging, but proper testing is not straightforward (comprehensive test environment is needed)

Deployment Venues

- Now using Debian packages to deliver development environment
- IOCs are synced with Git/Mercurial manually or with SNACK
- Base/modules built from source for advanced development cases
- New case: non-Debian systems (RedHat) for GPFS support
- How to keep existing capabilities (packaging, existing apps, deployment approaches etc.) while enabling new ones?

Containers



GitLab repo: Development Environment Dockerfile
EPICS
GitLab CI
docker
Build Host
Result delivered: Debian image with EPICS development packages
Tagged
docker-epics-env:latest
docker-epics-env:7be08d3172e7
docker-epics-env:b93fa09545a6
Base Dev Image
Base Dev Image
Base Dev Image
Central NFS
GitLab repo: IOC source and Dockerfile
EPICS
Cloned and built
docker
Engine
Run
IOC Image
Beamline IOC Host
Beamline IOC Host
. . .

- Sample setup, still many questions
- Base + major modules = 500MB image
- One image per IOC, or one image for all IOCs?
- Handling restarts, logs, troubleshooting?
- Handling persistence (autosave files)?
- Versioning and rollbacks?
- Manual or automatic delivery?
- How to do proper testing?
- How to make it developer-friendly?
- Easy to set up for one IOC, hard for entire facility

Current Concept Setup

- Dev Dockerfile – EPICS Dev Environment with Debian 9 as base
- GitLab CI – runner with a Shell executor on a build server
- DockerTools – convenience wrapper script to create and work with images and containers (used by the CI build)
- IOC Dockerfile – content-aware template for IOC image builds with Dev Environment image as base (adds softioc user, creates autosave folder, uses sysv-rc-softioc to install/enable/start IOC)
- Can easily deploy one IOC, scaling needs further consideration

Thank you