

# epics-containers

A workshop to look at containerizing EPICS IOCs

**EPICS Collaboration Meeting @ ORNL**

**Sept 16th, 2024**

giles knap

Beamline Controls – Diamond Light Source

# Agenda

- 13:10 – Presentation: an overview of epics-containers
- 13:50 – Questions
- 14:10 – Practical: experimenting with a simulation beamline
- 14:50 – Break
- 15:10 – Practical: introducing developer containers for generic IOCs
- 16:10 – Demo: Kubernetes and ArgoCD on a DLS test beamline
- 16:40 – Questions and discussion
- 17:00 - End

# Preamble

- This presentation is about how DLS will be building and managing IOCs using containers.
- However, we have applied the following principals to make this work useful to other facilities:
  - **Open source first.** All source, re-usable containers, example beamlines and documentation is available at <https://github.com/epics-containers>.
  - **Modular.** All parts of the framework are as far as practical independent – you may adopt just the features you find useful.
  - **Standard EPICS only.** We use the default EPICS build system, upstream versions of EPICS base and support modules.
  - **Standard Tools only.** The tools used in the framework are widely adopted FOSS only.
- The SPARC beamline at INFN-LNF in Rome is already using epics-containers in production

# What?



## Applying modern industry practices for software delivery to EPICS IOCs



**Containers:** Package IOC software and execute it in a lightweight virtual environment



**Kubernetes:** Centrally orchestrates all IOCs at a facility

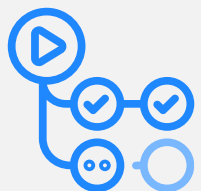
**Helm Charts:** Deploy IOCs into Kubernetes with version management



**Repositories:** Source and container repositories manage all the above assets

**CI:** Source repositories automatically build assets from source when updated

**CD:** Deployment repositories are automatically synced with the cluster



# Why?



- IOCs are decoupled from the OS
  - No modifying support modules to suit facility infrastructure
  - Allows us to use upstream support modules with no need for local forks
  - Develop and test anywhere
  - Protection from many security vulnerabilities
- Simple server setup: any Linux + container runtime only.
  - Very easy server OS upgrades
- Remove maintenance of internal management tools
- Kubernetes provides (not just for IOCs!):
  - A huge range of supporting tools both FOSS and licensed
  - Shared software deployment and management
  - Shared Logging, monitoring, alerting
  - Shared resource management: Disk / CPU / Memory
  - Skills required are transferrable
  - Just Google it when things go wrong

# How?

GitHub Repositories

epics-base repo

Dockerfile

ioc-aravis repo

Dockerfile

p47-services

bl47p-dcam-01

values.yaml

config/

ioc.yaml

bl47p-dcam-02

...

CI

FROM

CI

ec deploy bl47p-dcam-01 2024.9.1

epics-base image

/epics/  
epics-base/

ioc-aravis image

/epics/  
epics-base/  
support/  
ioc/

ioc-aravis container

/epics/  
epics-base/  
support/  
ioc/

/config/  
ioc.yaml

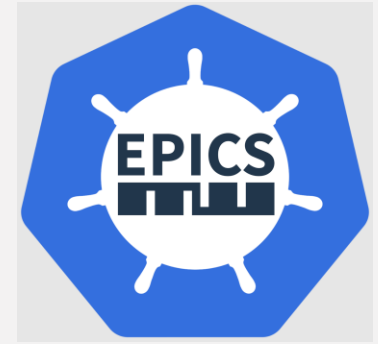
st.cmd  
ioc.db

start.sh

GitHub Container Registry

K8s worker

# Supporting Tools



- **ibek** – IOC builder for EPICS and Kubernetes
  - Runs inside the container at build time
    - Helpers for building Support and Generic IOC in the container environment
  - Runs inside the container at runtime startup
    - Makes IOC instance startup script and database from a YAML description
    - Extracts IOC instance engineering screens from the container



- **ec** – the epics containers CLI for developers
  - Runs outside the container
    - Helpers for building and deploy IOC Instances
    - Helpers for local debugging and testing of Generic IOCs
    - Thin wrapper around the tools git, helm, kubectl and argocd – can be used to learn these tools too



- **PVI** - Process Variables Interface
  - Provides structure for the PV interface to a device
  - Auto generates engineering screens for the device (bob/edm/adl)
  - Supplies DB metadata for use with Bluesky

# How to make an IOC Instance?

## 1. Create a beamline repository

- `copier copy gh:epics-containers/services-template-helm .`
- `cp -r services/.ioc-template services/my-new-ioc`
- Edit the new IOC Instance config in `services/my-new-ioc`

*1b. An **ibek** IOC YAML file listing the support 'entities' that we want to instantiate for this IOC instance.*

*1a. A helm chart values override file. The only required field is the URL of the Generic IOC to use.*

```
p47-services > services > bl47p-ea-dcam-01 > ! values.yaml > {} ioc-instance > image
1 ioc-instance:
2 | image: ghcr.io/epics-containers/ioc-adaravis-runtime:2024.8.1
3
```

```
p47-services > services > bl47p-ea-dcam-01 > config > ! ioc.yaml > [ ] entities
1 # yaml-language-server: $schema=https://github.com/epics-containers
2 ioc_name: bl47p-ea-dcam-01
3 description: GigE Sample camera for beamline p47
4
5 entities:
6 - type: epics.EpicsCaMaxArrayBytes
7   max_bytes: 3000000
8
9 - type: ADaravis.aravisCamera
10   CLASS: AVT_Mako_G234B
11   ID: 192.168.250.3
12   P: BL47P-EA-DET-01
13   PORT: DET.CAM
14   R: ":DET:"
```

## 2. Deploy the IOC instance:

- Tag and push the beamline repo
- `ec ioc deploy bl45p-ea-ioc-01 2024.9.1`



# How to make a Generic IOC?

## 1. Create a Generic IOC repo:

- copier copy gh:epics-containers/ioc-template
- edit Readme.md
- edit Dockerfile: add COPY and RUN for each support module using ibek-support recipes

## 2. Deploy Generic IOC to your container registry:

- Tag and push the generic IOC repo
- CI then:
  - Builds the container image
  - Publishes it to GHCR
  - Publishes a JSON schema for the ibek 'entities' provided inside the container

*Dockerfile: iocStats serves as an example for how to add additional support modules*

```
3 ARG BASE=7.0.8ec2
4 ARG REGISTRY=ghcr.io/epics-containers
5 ARG RUNTIME=${REGISTRY}/epics-base${IMAGE_EXT}-runtime:${BASE}
6 ARG DEVELOPER=${REGISTRY}/epics-base${IMAGE_EXT}-developer:${BASE}
7
8 ##### build stage #####
9 FROM ${DEVELOPER} AS developer
10
11 # The devcontainer mounts the project root to /epics/generic-source
12 # Using the same location here makes devcontainer/runtime differences transparent
13 ENV SOURCE_FOLDER=/epics/generic-source
14 # connect ioc source folder to its know location
15 RUN ln -s ${SOURCE_FOLDER}/ioc ${IOC}
16
17 # Get the current version of ibek
18 COPY requirements.txt requirements.txt
19 RUN pip install --upgrade -r requirements.txt
20
21 WORKDIR ${SOURCE_FOLDER}/ibek-support
22
23 # copy the global ibek files
24 COPY ibek-support/_global/ _global
25
26 COPY ibek-support/iocStats/ iocStats
27 RUN iocStats/install.sh 3.2.0
28
29 #####
30 # TODO - Add further support module installations here
31 #####
32
33 # get the ioc source and build it
34 COPY ioc ${SOURCE_FOLDER}/ioc
35 RUN cd ${IOC} && ./install.sh && make
36
37 # install runtime proxy for non-native builds
38 RUN bash ${IOC}/install_proxy.sh
39
40 ##### runtime preparation stage #####
```

# How to Add a new Support Module 1.

## 1. Add a folder in the ibek-support repo

- Shared ibek-support on GitHub
- Or private ibek-support per facility
- Public Generic IOCs should only use shared ibek-support
- Internal Generic IOCs may use both

## 2. Add a new install.sh file:

- Use the new support in your Generic IOC Dockerfile

*Install.sh example for ADSimDetector. Most install.sh would look almost identical to this but You can add custom steps as needed –it's just bash.*

```
ioc-template-example > ibek-support > ADSimDetector > $ install.sh
1  #!/bin/bash
2  #####
3  ##### install script for ADSimDetector Module #####
4  #####
5
6  # ARGUMENTS:
7  # $1 VERSION to install (must match repo tag)
8  VERSION=${1}
9  NAME=ADSimDetector
10 FOLDER=$(dirname $(readlink -f $0))
11
12 # log output and abort on failure
13 set -xe
14
15 # get the source and fix up the configure/RELEASE files
16 ibek support git-clone ${NAME} ${VERSION} --org http://github.com/areaDetector/
17 ibek support register ${NAME}
18
19 # declare the libs and DBDs that are required in ioc/iocApp/src/Makefile
20 ibek support add-libs simDetector
21 ibek support add-dbds simDetectorSupport.dbd
22
23 # global config settings
24 ${FOLDER}/../_global/install.sh ${NAME}
25
26 # compile the support module
27 ibek support compile ${NAME}
28 # prepare *.bob, *.pvi, *.ibek.support.yaml for access outside the container.
29 ibek support generate-links ${FOLDER}
30
```

# How to Add a new Support Module 2.

## 1. Add a folder in the ibek-support repo

- Shared ibek-support on GitHub
- Or private ibek-support per facility
- Public Generic IOCs should only use shared ibek-support
- Internal Generic IOCs may use both

## 2. Add a new install.sh file:

- Use the new support in your Generic IOC Dockerfile

## 3. Add an ibek support YAML description:

- This allows us to describe our IOC instances as ibek IOC YAML.

*Support YAML example for ADSimDetector. Declares instance arguments, startup script lines and database template substitutions.*

```
ioc-template-example > ibek-support > ADSimDetector > ! ADSimDetector.ibek.support.yaml > [ ] entity_models > {} 0 >
1  # yaml-language-server: $schema=https://github.com/epics-containers/ibek/
2  module: ADSimDetector
3
4  entity_models:
5    - name: simDetector
6      description: Creates a simulation detector
7      parameters:
8        P:
9          type: str
10         description: Device Prefix
11        R:
12          type: str
13         description: Device Suffix
14        PORT:
15          type: id
16         description: Port name for the detector
17        # ----- Other Parameters omitted for clarity -----
18
19      pre_init:
20        - type: text
21          value: |
22            # simDetectorConfig(portName, maxSizeX, maxSizeY, dataType, maxB
23            simDetectorConfig("#{PORT}", {{WIDTH}}, {{HEIGHT}}, {{DATATYPE
24
25      databases:
26        - file: $(ADSIMDETECTOR)/db/simDetector.template
27          args: { P, R, PORT, TIMEOUT, ADDR }
```

# Developer Containers



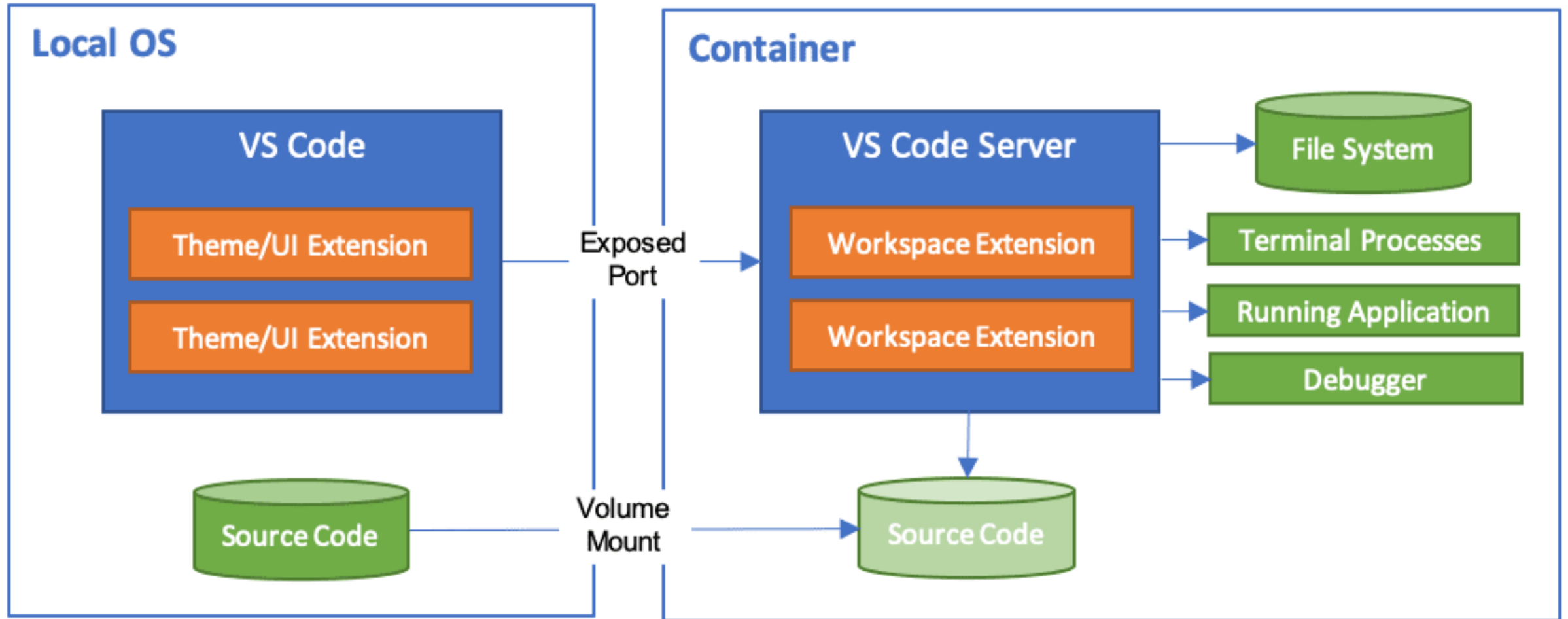
epics-containers development defines 3 levels of changes:

1. Changing IOC instance details only
  - Edit values.yaml or ioc.yaml in your beamline repository
  - Push and tag the changes, re-deploy the update instance
2. Changing a Generic IOC
  - Edit Dockerfile or ibek-support sub-module in a Generic IOC repository
  - Push changes to publish a new container image
  - Go to 1. to update affected instances
3. Changing Support Modules
  - Edit the support module, verify and push and tag source changes.
  - Go to 2. to update a Generic IOC to include the new support version

2 and 3 require rebuilding and deploying containers. For this reason, we use Developer Containers for a fast "inner loop".

For epics-containers the generic IOC container image is an ideal developer container.

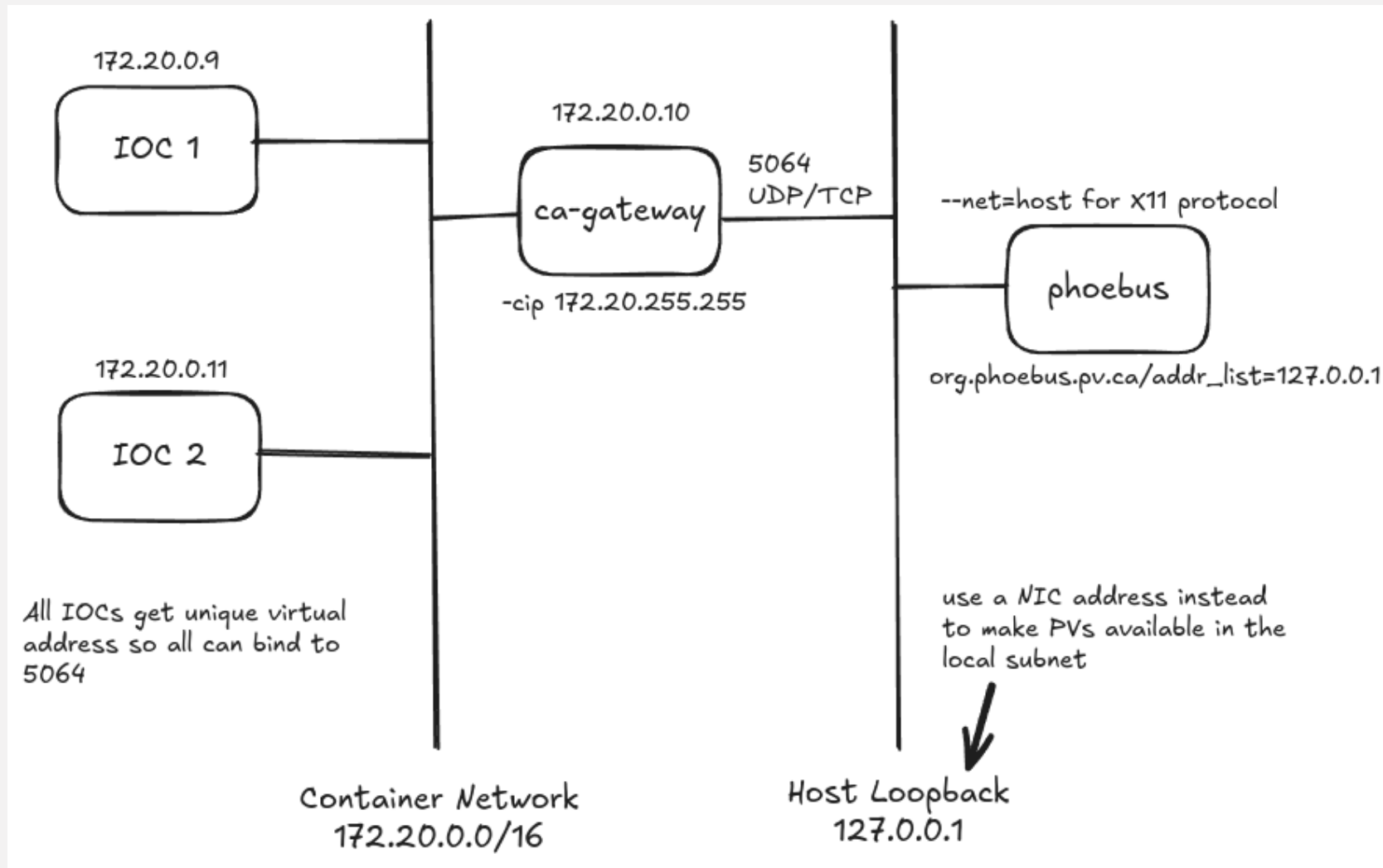
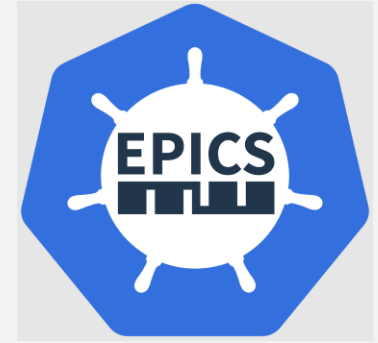
# Developer Containers



See developer target in <https://github.com/epics-containers/ioc-template/blob/main/template/Dockerfile>

# Networking

For local development and testing we configure the network like this:



# Networking

## network=host



At DLS we use network=host for Kubernetes IOCs

- This means IOCs run without network isolation and look exactly like traditional IOCs from the client's perspective

### Motivation

- Channel Access cannot route into the container network
- We did not want to pass all PVs through a ca-gateway
- Channel Access is not the only protocol that will not route into CNI
- For example, GigE streaming protocol will also fail
- Any protocol that does not like NAT will have this problem

# Current Status - Sept 2024

- The epics-containers framework is ready for wider exposure and feedback
- There are tutorials to get people started using the framework
- There are a growing number of reference generic IOCs
- Example beamline repositories are also available
- DLS has several beamline clusters running a handful of containerized IOCs in production, plus some fully containerized test beamlines.
- DLS aims to have a representative set of production beamlines fully containerized before the DII dark period 12/2027.
- And all beamlines fully containerized at the end of the dark period 06/2029.



# One Take Away



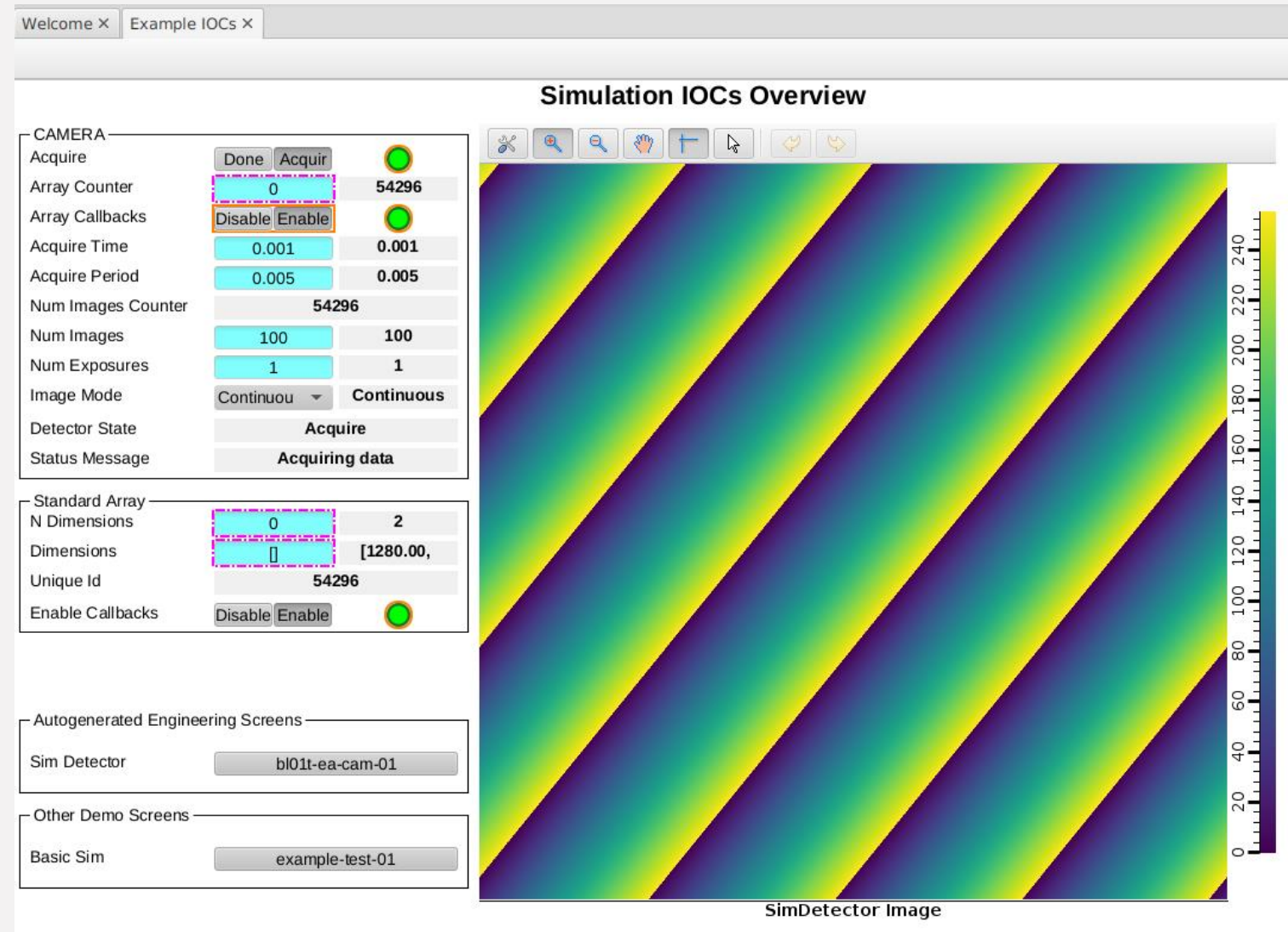
<https://github.com/epics-containers>

- Includes:
  - Tutorials
  - Documentation
  - Templates
  - Source code
  - A small but growing number of Generic IOC images
  - Example beamlines
  - A Simulation beamline

Questions ?

# Hands On: Simulation Beamline

- Includes
  - Motion IOC
  - AreaDetector IOC
  - Ca Gateway
  - Phoebus in a container
  - Screens to for the IOCs
- All launched by docker compose



# Hands On: Simulation Beamline

- Install git and docker / podman + docker compose
  - [https://epics-containers.github.io/main/tutorials/setup\\_workstation.html](https://epics-containers.github.io/main/tutorials/setup_workstation.html)
- Launch a simulation beamline!
  - git clone <https://github.com/epics-containers/example-services.git>
  - cd example-services.git
  - source ./environment.sh
  - ec up -d
- See README.md for more commands
- Examine the configuration
  - compose.yaml
  - services/\*/config/\*
  - See docker compose docs: <https://docs.docker.com/reference/compose-file/>

# Hands On: Make your own beamline

- <https://epics-containers.github.io>
- Try
  - Create a Beamline Repository
  - Deploying and Managing IOC Instances
  - Create an IOC Instance

If you don't have time to complete these tutorials today, you can do them afterwards at your leisure.

You can open an issue against the documentation project or contact the authors if you need help or have any ideas to discuss.

Also see <https://github.com/orgs/epics-containers/discussions>

# Hands On: Developer Container Tutorial

- Docker users:
  - Add this to `$HOME/.profile` and `logout/in`
    - `export EC_REMOTE_USER=$USER`
- <https://epics-containers.github.io>
- Try
  - Developer Containers
  - Developer Containers Part 2
  - Changing a Generic IOC

# Kubernetes

- Kubernetes is by far the dominant container orchestration platform today
- Open-sourced by Google in 2014
- Managed by the Cloud Native Computing Foundation, part of the Linux Foundation
- CNCF looks after a large list of open-source applications that run in Kubernetes
- At DLS we will have a Kubernetes Cluster per beamline, one for the accelerator and a large central cluster for centralized services.

# Demo: a Kubernetes Beamline at DLS

- p47 is a training beamline with 2 detectors, 2 motors, 1 pandabox
- Each Kubernetes Clusters at DLS runs standard services including:-
  - A landing page to access all user services
  - Kubernetes Dashboard – manage resources in the cluster
  - Alert Manager – sets thresholds and configures recipients of alerts
  - Prometheus – monitoring with time series Database
  - Grafana – rich visualization of the above data
  - StacksRox – monitor running containers for Common Vulnerabilities and Exploits
  - Keycloak – single sign on authorization service
  - Kynervo – policy engine
  - Argo CD – declarative GitOps continuous deployment



Questions ?

# Remaining Slides

- Images of demo screens in case I can't connect to DLS
- Some overview diagrams for discussion if needed

# Demo: a Kubernetes Beamline at DLS

## P47

**P47 Beamline Synoptic**

X Stage Position: -18.000 mm -18.000 mm  
Stop: Stop  
Tweak Forward: Tweak Forward  
Tweak Step: 1.000 mm  
Tweak Reverse: Tweak Reverse

Rotation Stage Position: 703.998 mm 703.998 mm  
Stop: Stop  
Tweak Forward: Tweak Forward  
Tweak Step: 18.000 mm  
Tweak Reverse: Tweak Reverse

Sample Camera  
Num Queued Arrays: 0  
Wait For Plugins: No Yes  
Acquire: Done Acquir ☒  
Array Counter: 0 2557  
Array Callbacks: Disable Enable ☒  
Acquire Time: 0.015 0.015  
Acquire Period: 0.031 0.031  
Num Images Counter: 2556  
Num Images: 1 1  
Num Exposures: 1 1  
Image Mode: Continuous Continuous  
Trigger Mode: Off Off  
Detector State: Acquire  
Status Message:

Overview Camera  
Acquire: Done Acquir ☒  
Array Counter: 154

Auto Generated Engineering Screens  
Overview Camera: BL47P-EA-DET-02  
Sample Camera: BL47P-EA-DET-01  
Motors: BL47P-MO-MAP-01  
Panda: BL47P-MO-PANDA-01

**Sample**

**Overview**

**ADAravis Camera**

AD Aravis  
AR Frames Completed: 1820  
AR Frame Failures: 0  
AR Frame Underruns: 0  
AR Missing Packets: 0  
AR Resent Packets: 0  
AR Packet Resend: Always  
AR Packet Timeout: 20000 us  
AR Frame Retention: 100000 us  
AR Reset Camera: 0  
AR Convert Pixel: Mono16L Mono16Low  
AR Shift Dir: None None  
AR Shift Bits: 4 4

AD Shutter  
Shutter Mode: None None  
Shutter Status: ☒  
Shutter Control: Close Open ☒  
Shutter Open Delay: 0.000 0.000  
Shutter Close Delay: 0.000 0.000

AD Collect  
Num Queued Arrays: 0  
Wait For Plugins: No Yes  
Acquire: Done Acquir ☒  
Array Counter: 0 1820  
Array Callbacks: Disable Enable ☒  
Acquire Time: 0.500 0.500  
Acquire Period: 0.004 0.004  
Num Images Counter: 1820  
Num Images: 1 1  
Num Exposures: 1 1  
Image Mode: Continuous Continuous  
Trigger Mode: Off Off  
Detector State: Acquire  
Status Message: 0.000  
Time Remaining: 0.000

AD Setup  
Port Name: DET2.CAM  
Manufacturer: Allied Vision Technologies  
Model: Mako G-040C  
Serial Number: 50-0536985400  
Firmware Version: 00.01.54.21000  
SDK Version: 0.8.31  
Driver Version: 2.3  
AD Core Version: 3.12.1

AD Attr File  
ND Attributes File:   
ND Attributes Macros:   
ND Attributes Status: File not found

asynNDArrayDriver Ungrouped  
Empty Free List:   
Acquire Busy CB: ☒  
Bayer Pattern: RGGB  
Array Size Z: 0  
Codec:   
Compressed Size: 0  
Unique Id: 0  
Time Stamp: 0.000  
Epics TS Sec: 0  
Epics TS Nsec: 0  
Pool Max Mem: 0.0 MB  
Pool Used Mem: 9.1 MB  
Pool Alloc Buffers: 24  
Pool Free Buffers: 1  
N Dimensions: 0 0  
Dimensions: 0 0

AD Readout  
Array Size X: 728  
Array Size Y: 544  
Array Size: 396032  
Data Type: UInt8 UInt8  
Color Mode: RGB1 RGB1  
Max Size X: 728  
Max Size Y: 544  
Bin X: 1 1  
Bin Y: 1 1  
Min Y: 0 0  
Min X: 0 0  
Reverse X: No Yes ☒  
Reverse Y: No Yes ☒  
Size X: 728 728  
Size Y: 544 544  
Gain: 0.000 0.000

AD Connect Camera: 1

ADDriver Ungrouped  
Read Status:   
Num Exposures: 0  
String To Server: <not used by driver>  
String From Server: <not used by driver>  
Shutter Control EPICS: ☒  
Temperature Actual: 54.1 C  
Frame Type: Normal Normal  
Temperature: 25.0 C 25.0 C

# Cluster landing page

Welcome to the Pollux Kubernetes Cluster landing page

## Pollux Grafana

Grafana instance for monitoring the Pollux cluster

## Pollux Prometheus

Prometheus instance for monitoring the Pollux Cluster

## Pollux Alertmanager

Prometheus alertmanager instance for monitoring the Pollux cluster

## Pollux K8s Dashboard

Kubernetes Dashboard for the Pollux cluster

## Kubernetes User Guide

Dev Portal user guide for Kubernetes

## Jupyterhub Test

Testing instance of jupyterhub

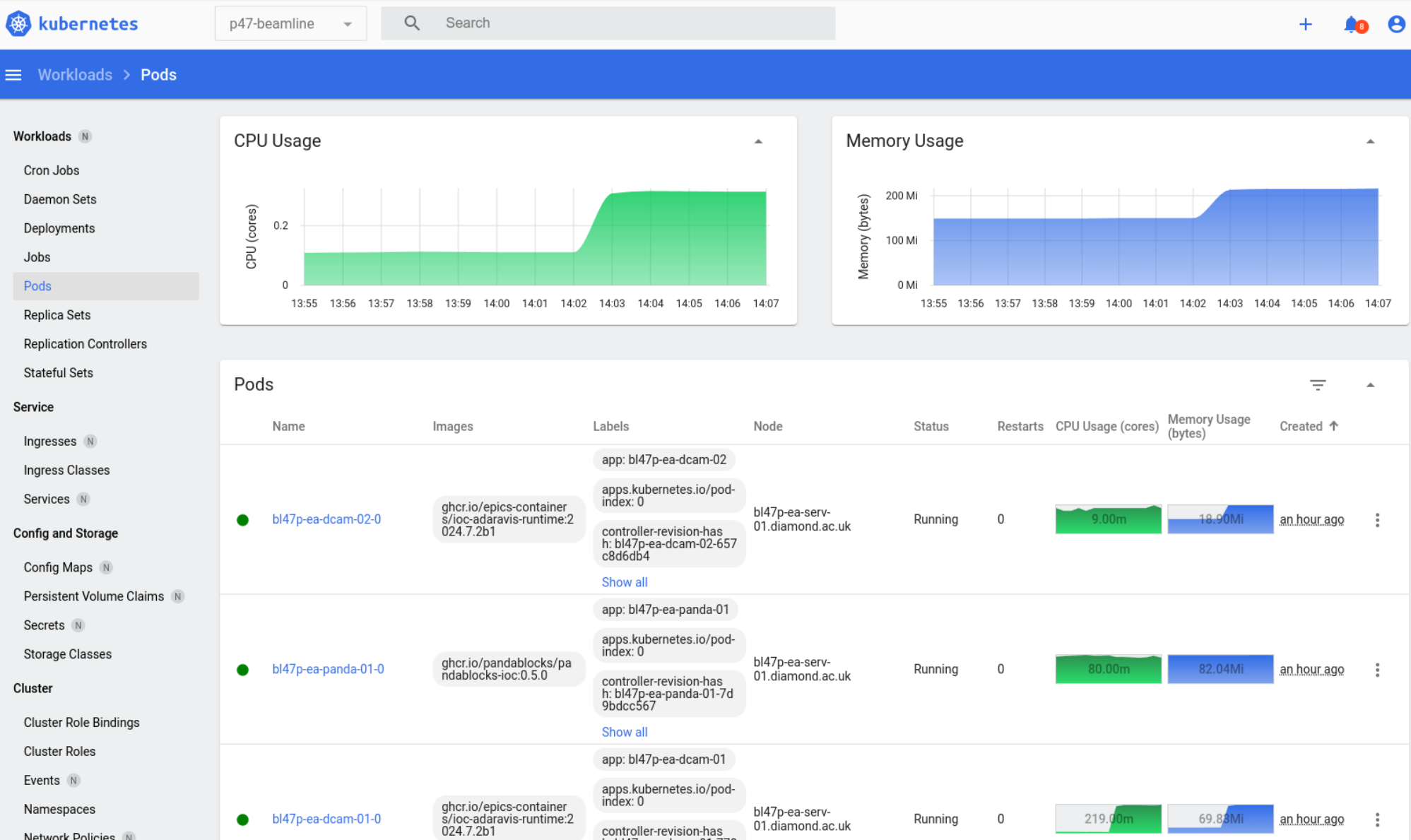
## Pollux KeyCloak

Keycloak instance for the Pollux cluster

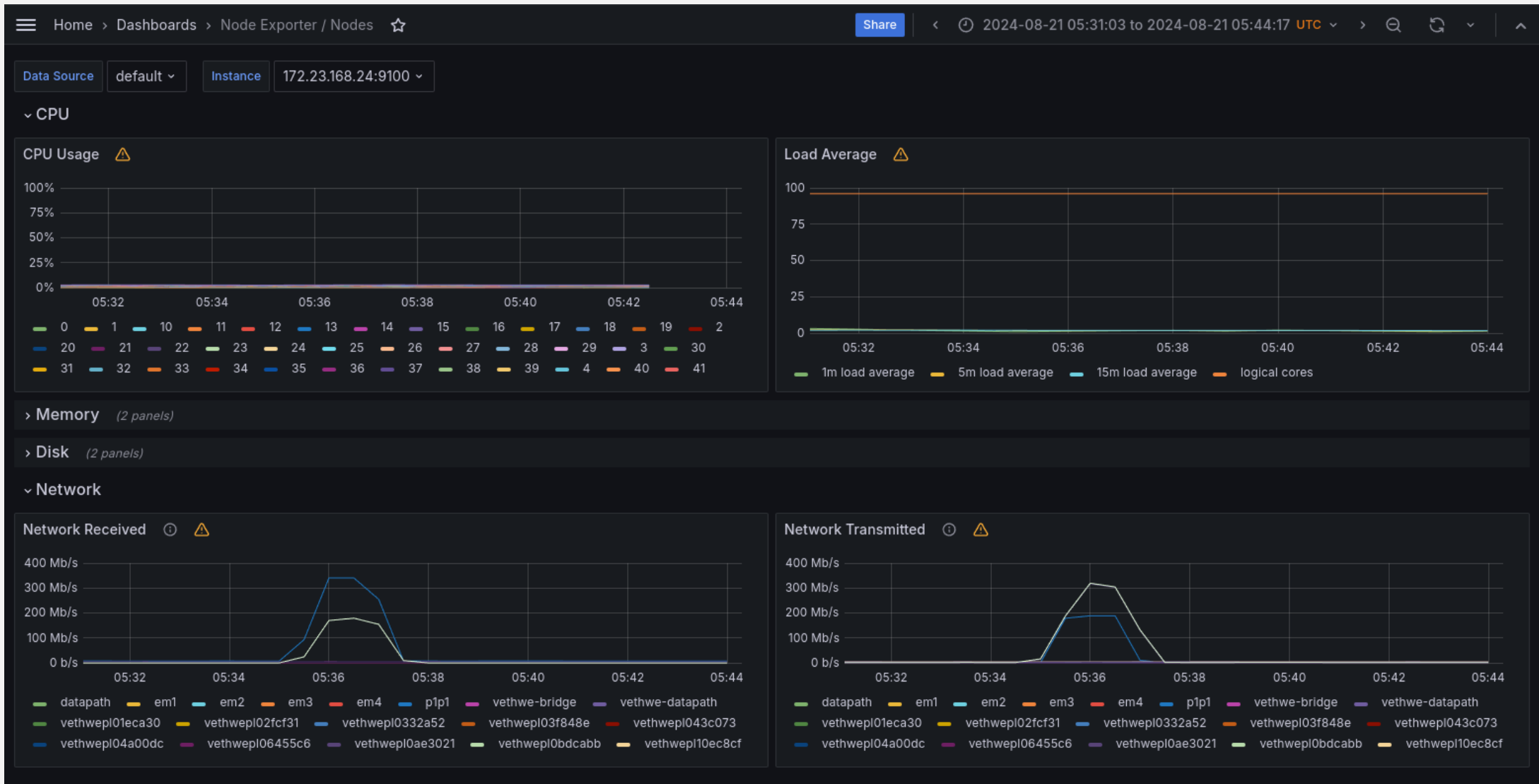
## Pollux Stackrox

Stackrox security dashboard for the Pollux cluster

# Kubernetes Dashboard



# Grafana



StackRox

CLI

?

--

Dashboard

1 Cluster

13 Nodes

88 Violations

19 Deployments

13 Images

117 Secrets

Last updated 22/08/2024 at 14:08

Network

Violations

Vulnerability Management (2.0)

Vulnerability Management (1.0)

Configuration Management

Risk

Platform Configuration

Dashboard

Review security metrics across all or select resources

Resources: All clusters All namespaces

88 policy violations by severity

View all

44 Low

32 Medium

9 High

3 Critical

Most recent violations with critical severity

Rapid Reset: Denial of Service...

epics-opis

08/21/2024 | 12:15:49PM

Rapid Reset: Denial of Service...

epics-opis

07/25/2024 | 8:22:20AM

Rapid Reset: Denial of Service...

daq-rabbitmq

07/25/2024 | 8:22:15AM

Images at most risk

View all

Images	Risk priority	Critical CVEs	Important CVEs
sonatype/nexus3	1	7 fixable	36 fixable
bitnami/rabbitmq	4	0 fixable	9 fixable
library/nginx	8	1 fixable	0 fixable
diamond-privreg... ile-converter	15	2 fixable	7 fixable
diamondlightsource/blueapi	17	0 fixable	1 fixable
pandablocks/pandablocks-ioc	29	0 fixable	0 fixable

Deployments at most risk

View all

Deployment	Resource location	Risk priority
daq-rabbitmq	in "pollux / bl45p"	5
epics-opis	in "pollux / bl45p"	16
epics-opis	in "pollux / p47-beamline"	16
nrmb-sonatype-nexu...manager	in "pollux / sonatype-nexus-repository-manager"	23
daq-blueapi	in "pollux / bl45p"	45

11 Aging images

View all

Aging Period	Image Count
30-90 days	5
90-180 days	2
180-365 days	1
>1 year	3

opped 0 overruns 0 carrier 0 collisions 0 /home/hqv27681/.config/containers/containers.c

# Argo CD for p47-beamline

Applications

APPLICATIONS TILES



+ NEW APP

SYNC APPS

REFRESH APPS



Search applications...

Sort: name ▼ Items per page: 20 ▼

**p47-beamline/bl47p-ea-dcam-01**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/bl47p-ea-dcam-01

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/22/2024 12:18:11 (2 hours ago)

SYNC



REFRESH

DELETE

**p47-beamline/bl47p-ea-dcam-02**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/bl47p-ea-dcam-02

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/22/2024 12:21:02 (2 hours ago)

SYNC



REFRESH

DELETE

**p47-beamline/bl47p-ea-panda-01**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/bl47p-ea-panda-01

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/22/2024 12:18:12 (2 hours ago)

SYNC



REFRESH

DELETE

**p47-beamline/bl47p-ea-test-01**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/bl47p-ea-test-01

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/22/2024 12:18:09 (2 hours ago)

SYNC



REFRESH

DELETE

**p47-beamline/bl47p-mo-ioc-01**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/bl47p-mo-ioc-01

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/22/2024 12:18:12 (2 hours ago)

SYNC



REFRESH

DELETE

**p47-beamline/epics-opis**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/epics-opis

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/21/2024 13:08:13 (a day ago)

SYNC



REFRESH

DELETE

**p47-beamline/epics-pvcs**

Project: p47-beamline

Labels: argocd.argoproj.io/instance=p47-beamlin...

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... 2024.8.4

Path: services/epics-pvcs

Destinati... pollux

Namespa... p47-beamline



Created At: 08/21/2024 12:15:28 (a day ago)

Last Sync: 08/21/2024 12:15:45 (a day ago)

SYNC



REFRESH

DELETE

**p47-beamline/p47**

Project: p47-beamline

Labels:

Status:  Healthy  Synced

Repositor... <https://github.com/epics-containers/p47-...>

Target Re... main

Path: apps

Destinati... argus

Namespa... p47-beamline

Created At: 08/21/2024 12:15:22 (a day ago)

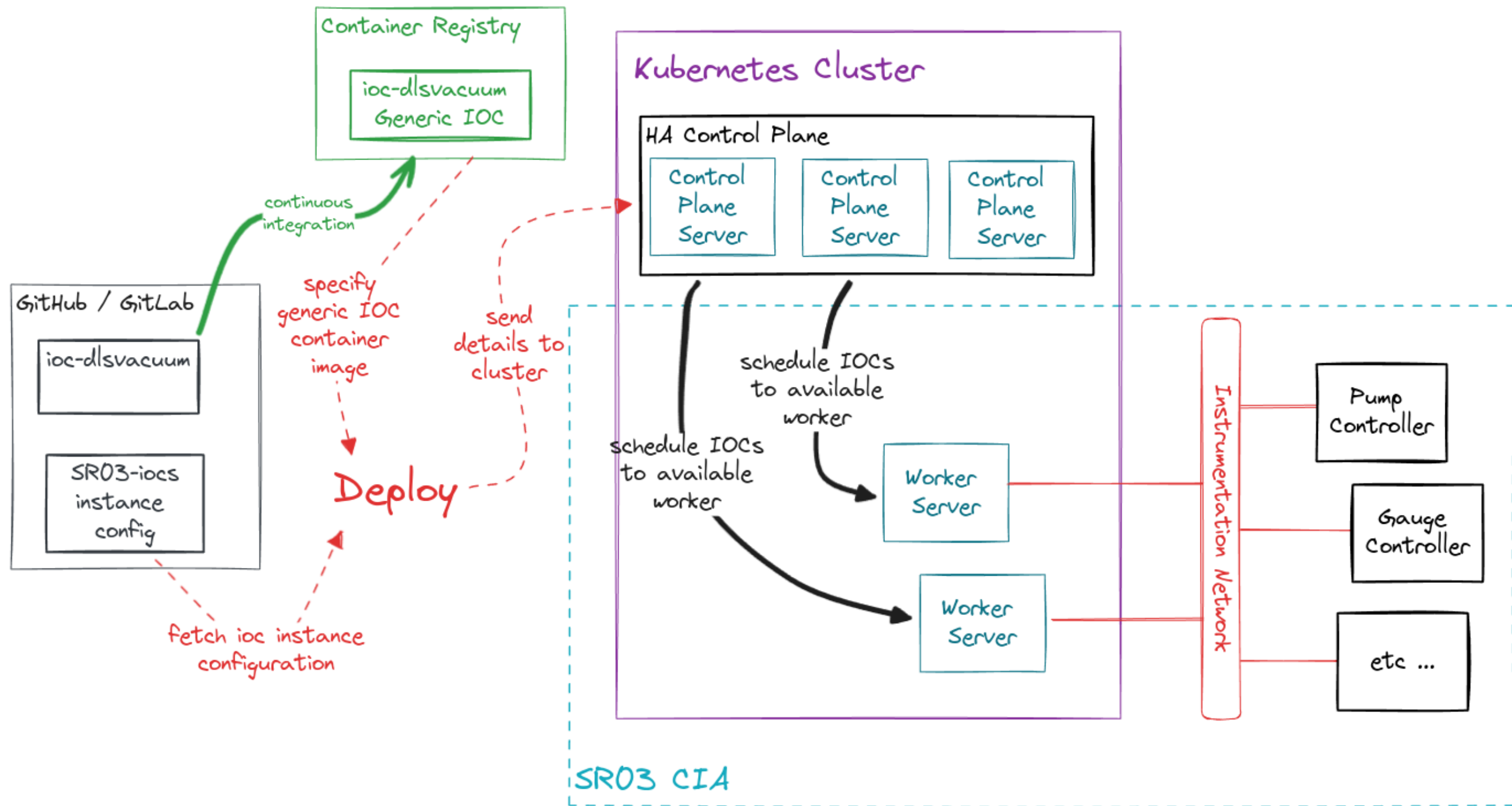
Last Sync: 08/21/2024 12:20:14 (a day ago)

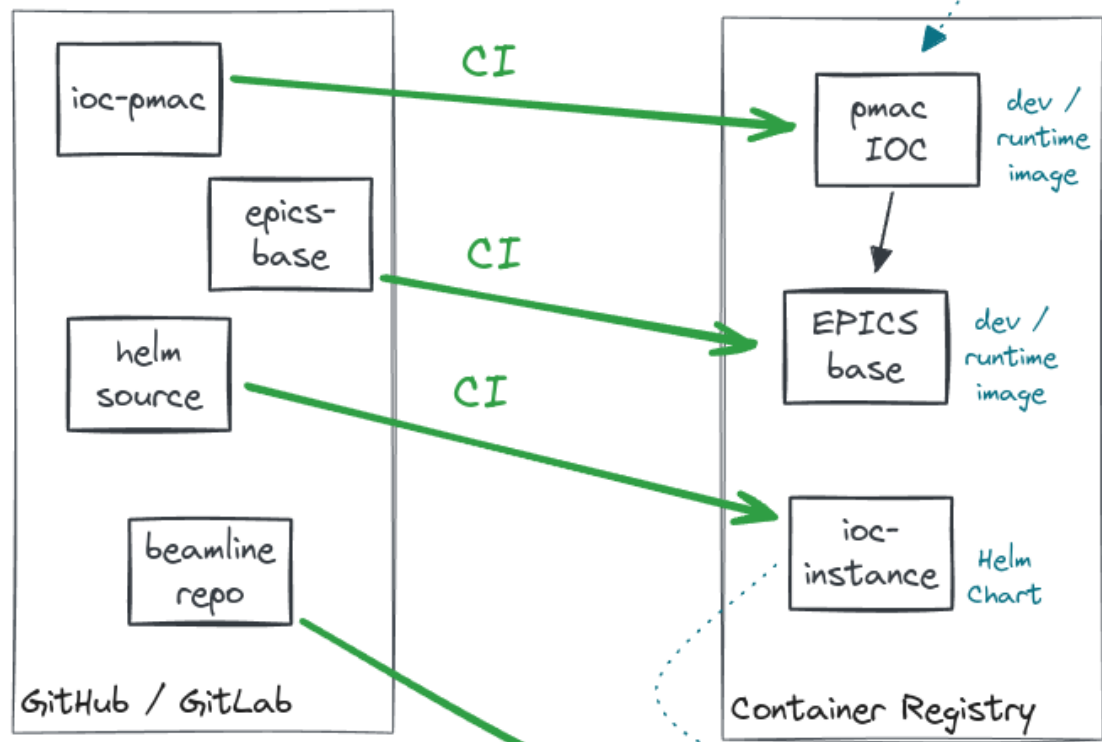
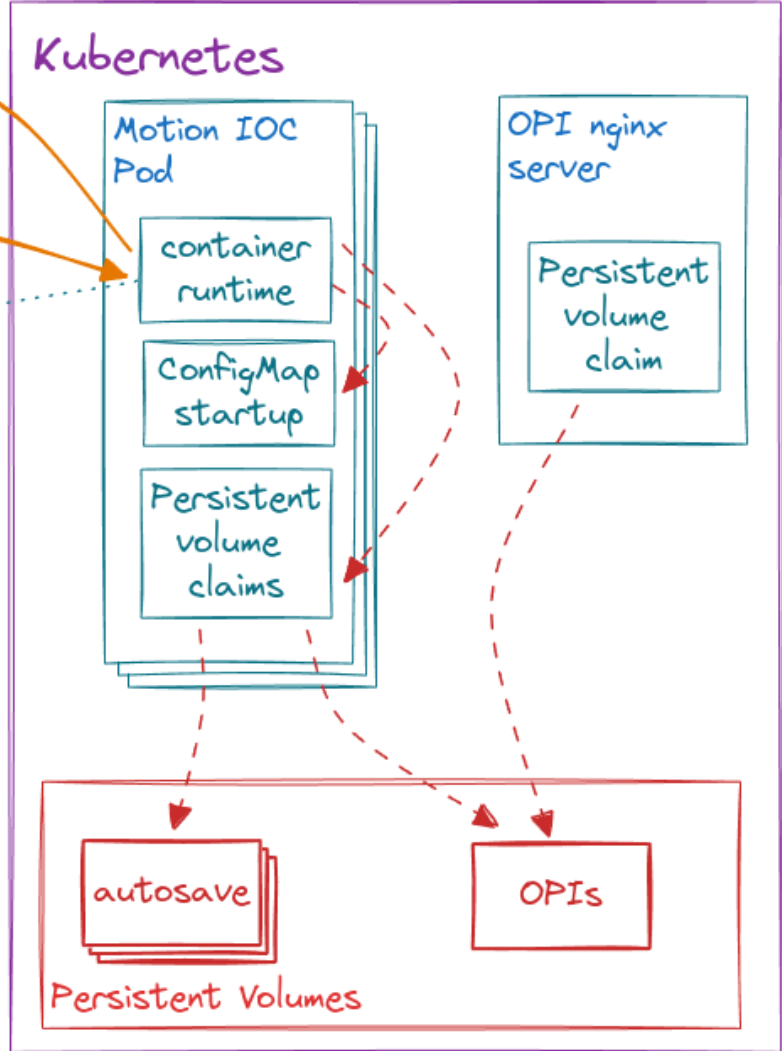
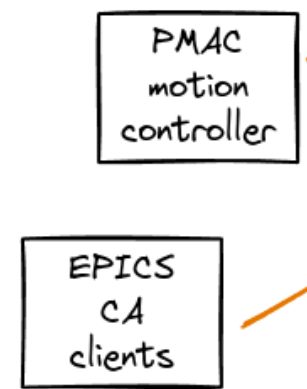
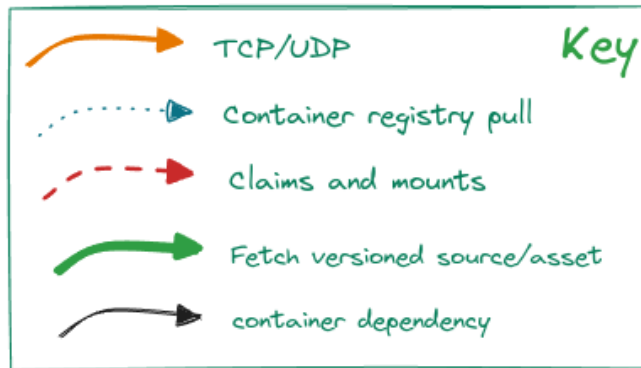
SYNC

REFRESH

DELETE







ec deploy bl45p-mo-ioc-01 2024.2.1