

UPDATING THE 14UD CONTROL SYSTEM AT HIAF (ANU)

From one VM and one iocApp to Gitlab CI/CD,
dockerized iocApps, (stateless) IOC's, and
(stateless) micro services: work in progress



Australian
National
University





Battisson, Stephen
Cooper, Alan
Heighway, Justin
Kafer, Chris
Kitchen, Tom
Linardakis, Peter
Lobanov, Nikolai
Tempra, Daniel
Tranter, Ben
Tranter, Ross
Tunningley, Thomas

Acknowledgements



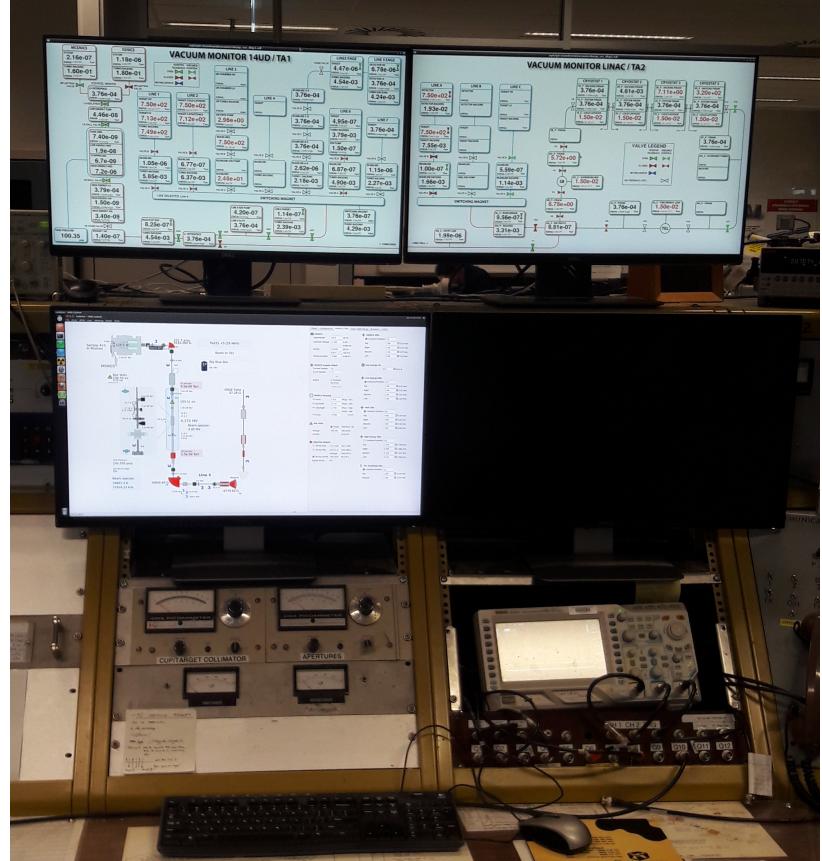
Introduction

- The ANU operates an NEC 14UD accelerator.
- The accelerator was built starting in mid 1971, and had first volts applied in February of 1973.
- The 14UD was the first of its kind.
- Built in an analog age,



Introduction

- The ANU operates an NEC 14UD accelerator.
- The accelerator was built starting in mid 1971, and had first volts applied in February of 1973.
- The 14UD was the first of its kind.
- Built in an analog age, mostly digital today
- Current state
 - Server: Ubuntu 12 VM
 - (DHCP, DNS, TFTP, NBD, LTSP)
 - EPICS 3.14.? and modules (-dirty)
 - OPI's written in python2 (wx, svg, ...)
 - running on LTPS clients (Ubuntu 12)
 - Special hardware: MVME crates (RTEMS 4.10), arm based IO's (oabi)



How should a control system look like?

- OPI's:
- EPICS IOC's
- Providing/Serving OPI's, IOC's and other services.



How should a control system look like?

- OPI's:
- EPICS IOC's
- Providing/Serving OPI's, IOC's and other services.

Bad user interface:

```
#include <iostream>

int main(){
    int selected_beamline=5;

    AcceleratorConfig* accelerator=new AcceleratorConfig(&selected_beamline);

    ...
    ...

    return 0;
}
```



How should a control system look like?

- OPI's:
- EPICS IOC's
- Providing/Serving OPI's, IOC's and other services.

Bad user interface:

```
#include <iostream>

int main(){
    int selected_beamline=5;

    AcceleratorConfig* accelerator=new AcceleratorConfig(&selected_beamline);

    ...
    ...

    return 0;
}
```

Good user interface:

```
#include <iostream>
#include <fstream>

int main(){
    ifstream fin("accelerator.cfg");

    AcceleratorConfig* accelerator=parseCfg(&fin);

    ...
    ...

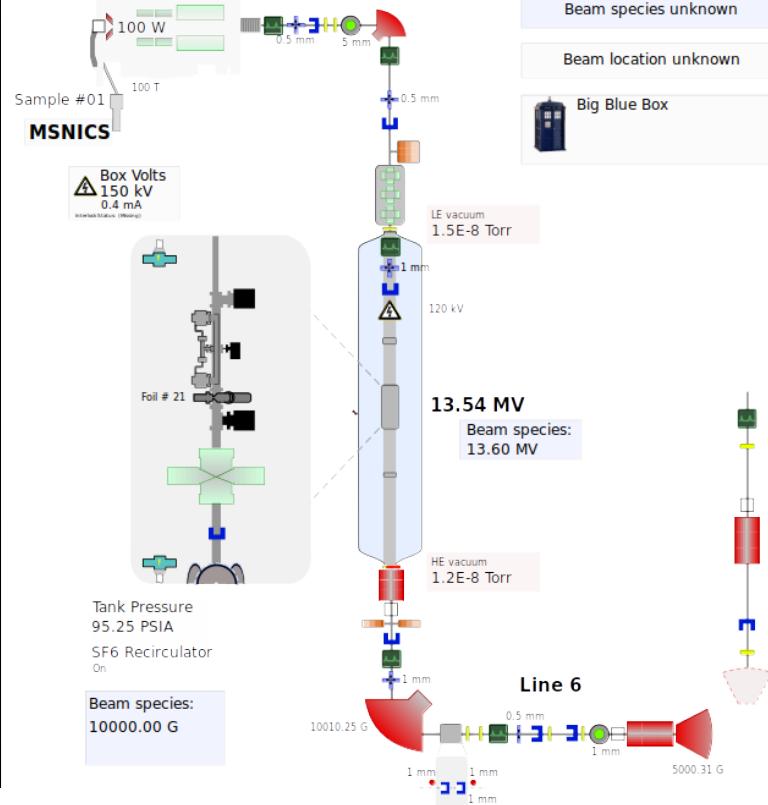
    return 0;
}
```



How should a control system look like?

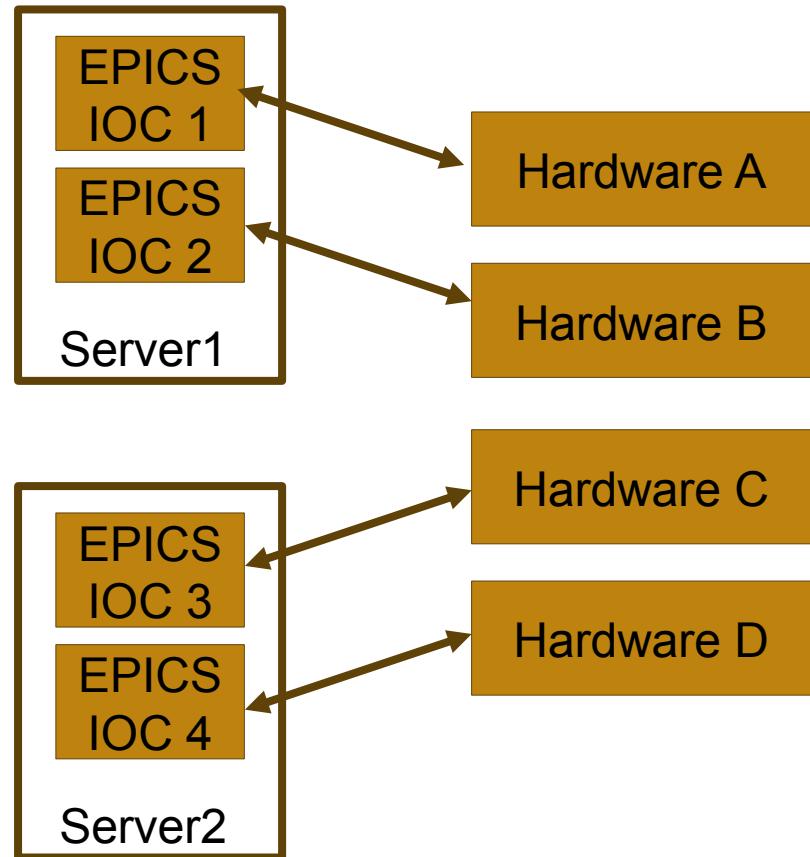
- OPI's:
 - Intuitive to use
 - Easy to develop and maintain (WYSIWYG)
- EPICS IOC's
- Providing/Serving OPI's, IOC's and other services.

Better user interface:



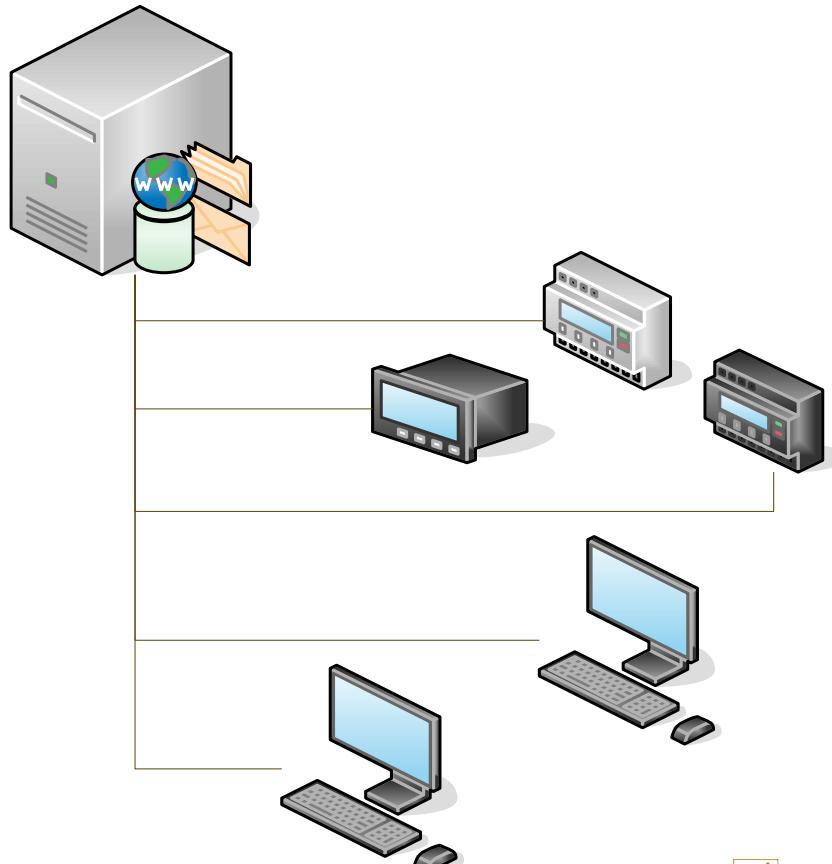
How should a control system look like?

- OPI's:
 - Intuitive to use -- I'm out --
 - Easy to develop and maintain (WYSIWYG)
- **EPICS IOC's**
 - Hardware agnostic where possible
 - Stateless (exception autosave)
 - Flexible: mix and match base+modules.
 - Stable: a change to base or a module shouldn't randomly break IOC's.
- Providing/Serving OPI's, IOC's and other services.



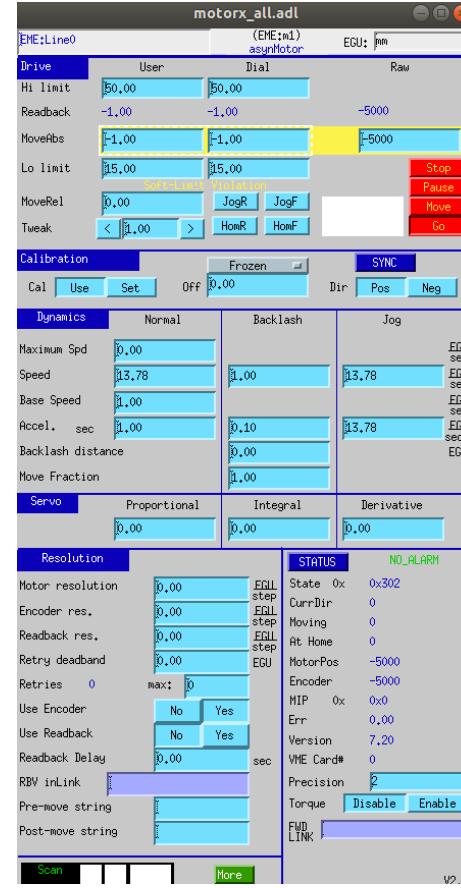
How should a control system look like?

- OPI's:
 - Intuitive to use -- I'm out --
 - Easy to develop and maintain (WYSIWYG)
- EPICS IOC's
 - Hardware agnostic where possible
 - Stateless (exception autosave)
 - Flexible: mix and match base+modules.
 - Stable: a change to base or a module shouldn't randomly break IOC's.
- **Providing/Serving OPI's, IOC's and other services.**
 - Failure tolerant with little or no down time
 - Easy to set up and maintain
 - Flexible: Move between hardware/servers, add services, ...



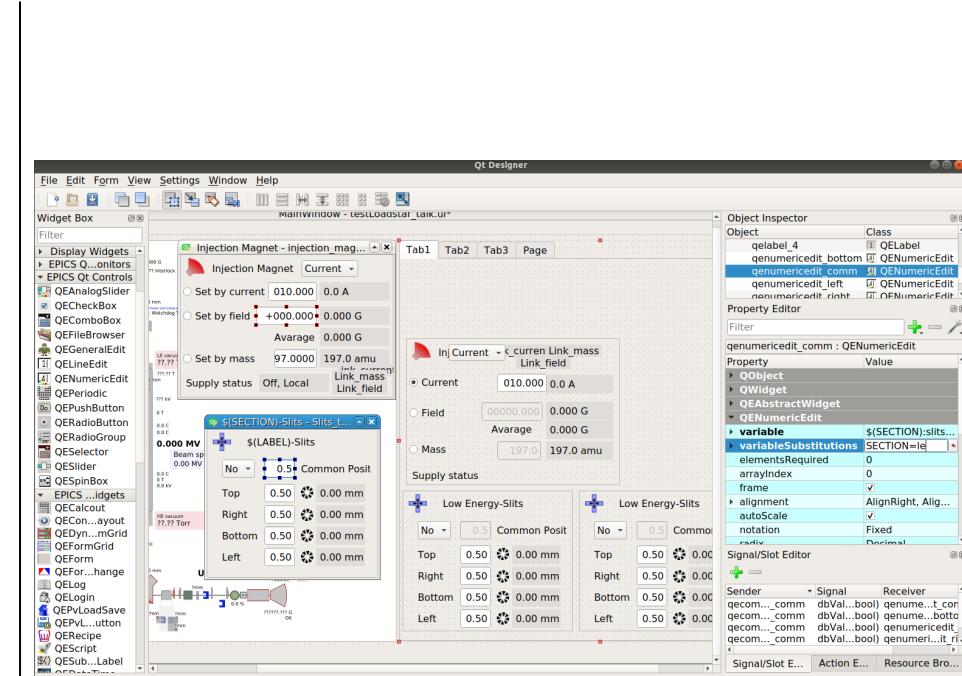
What to use for OPI's

- Available OPI's:
 - MEDM, EDM, qegui, CSS, ...
 - React-Automation-Studio, WebOPI, ...



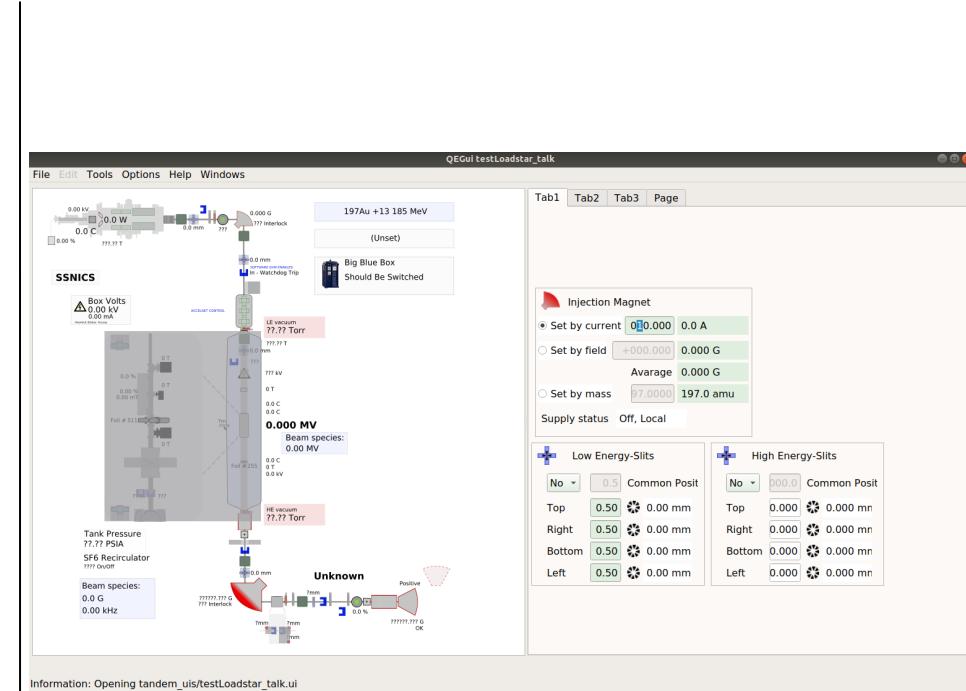
What to use for OPI's

- Available OPI's:
 - MEDM, EDM, qegui, CSS, ...
 - React-Automation-Studio, WebOPI, ...
- qegui
 - WYSIWYG
 - Allows for reusable sub ui's
 - SVG canvas, dynamic tabs: work in progress
 - Animation/Simulation of analog meters: work in progress



What to use for OPI's

- Available OPI's:
 - MEDM, EDM, qegui, CSS, ...
 - React-Automation-Studio, WebOPI, ...
- qegui
 - WYSIWYG
 - Allows for reusable sub ui's
 - SVG canvas, dynamic tabs: work in progress
 - Animation/Simulation of analog meters: work in progress



EPICS IOC's

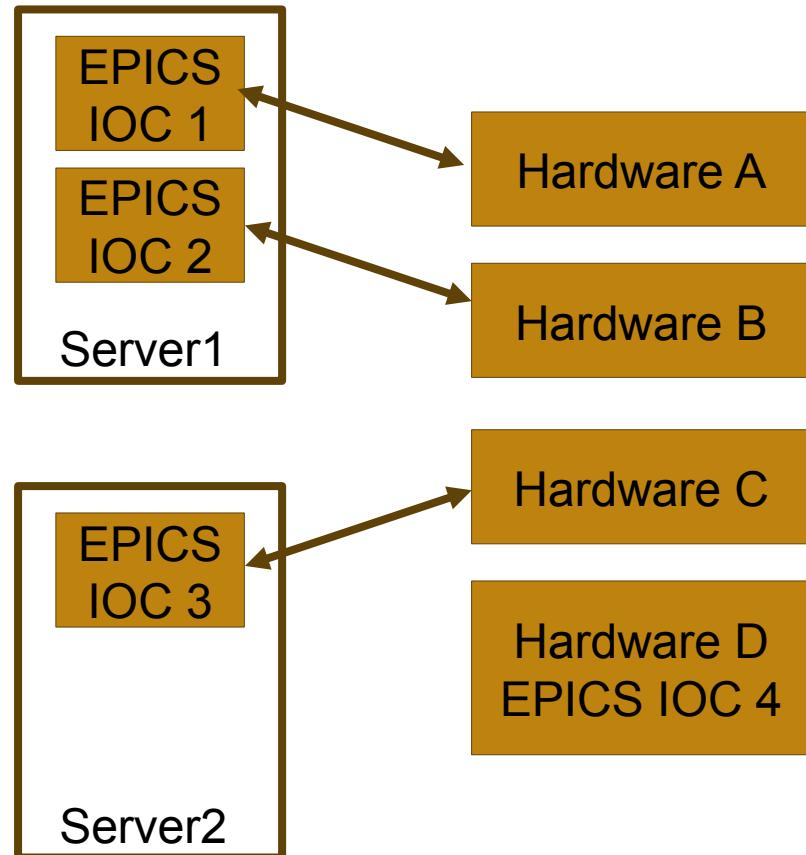
- **Hardware agnostic where possible**
 - Stateless (exception autosave)
 - Flexible: mix and match base+modules.
 - Stable: a change to base or a module shouldn't randomly break IOC's.



Using Ethernet enabled field devices, e.g.

MODBUS-TCP/IP Serial-to-Ethernet converter

3



EPICS IOC's

- **Hardware agnostic where possible**
- Stateless (exception autosave)
- Flexible: mix and match base+modules.
- Stable: a change to base or a module shouldn't randomly break IOC's.



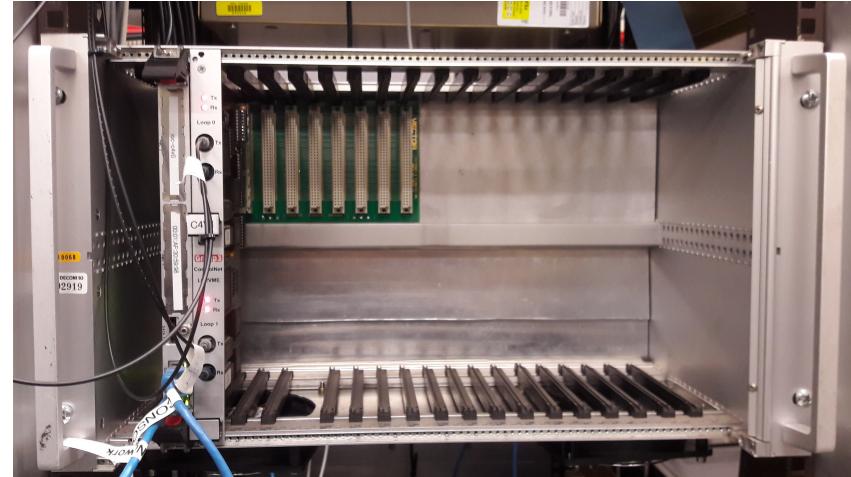
Using Ethernet enabled field devices, e.g.

MODBUS-TCP/IP
Serial-to-Ethernet converter

...

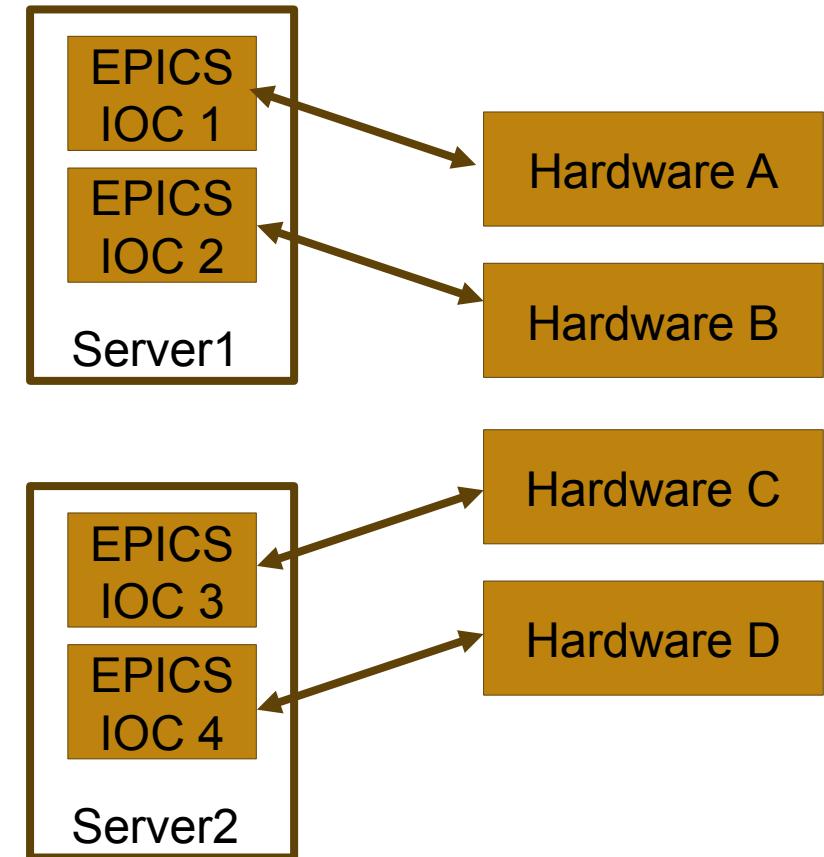
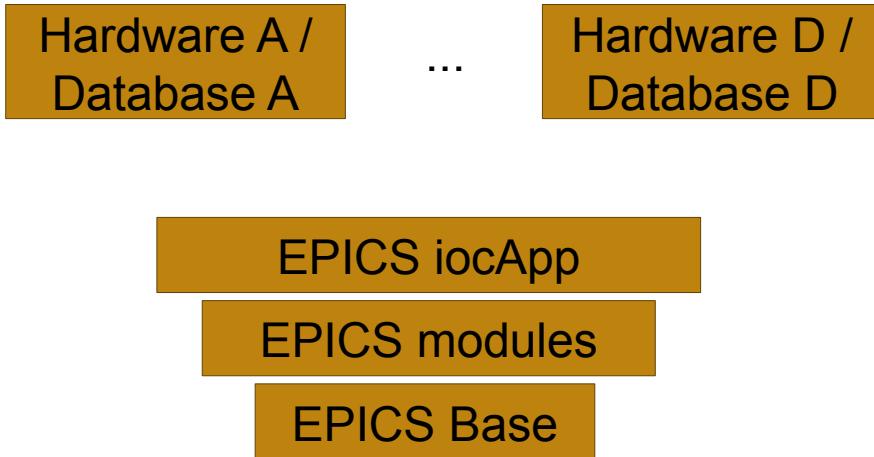
MVME crates and RTEMS

- EPICS R7 compiles against RTEMS 4.10, but
 - TFTP filesystem seems to be broken
 - Only NFSv2 implemented
- Our MVME crates are not supported by RTEMS 5 yet.



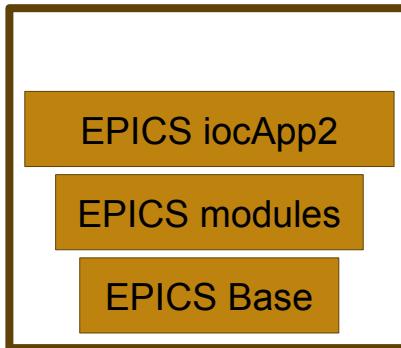
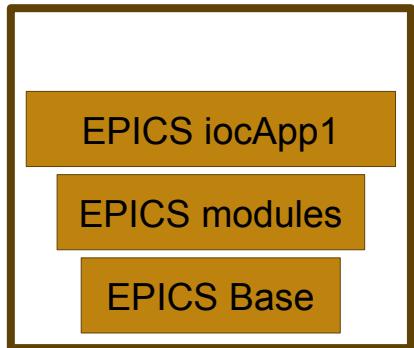
EPICS IOC's

- Hardware agnostic where possible
- Stateless (exception autosave)
- **Flexible: mix and match base+modules.**
- **Stable: a change to base or a module shouldn't randomly break IOC's.**



EPICS IOC's

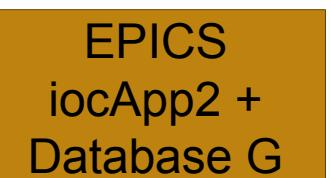
- Hardware agnostic where possible
- Stateless (exception autosave)
- **Flexible: mix and match base+modules.**
- **Stable: a change to base or a module shouldn't randomly break IOC's.**



iocApps deployed to production



...

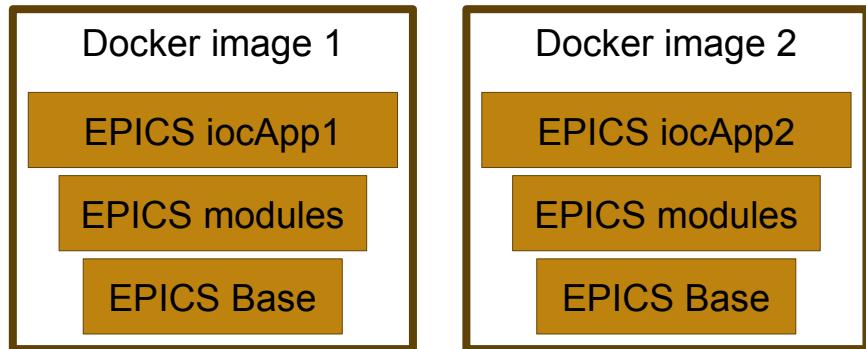


...



EPICS IOC's

- Hardware agnostic where possible
- Stateless (exception autosave)
- **Flexible: mix and match base+modules.**
- **Stable: a change to base or a module shouldn't randomly break IOC's.**



Containers deployed to production

Docker image 1
+ Database A

Docker image 1
+ Database D

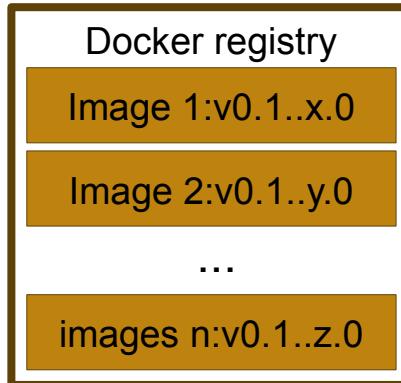
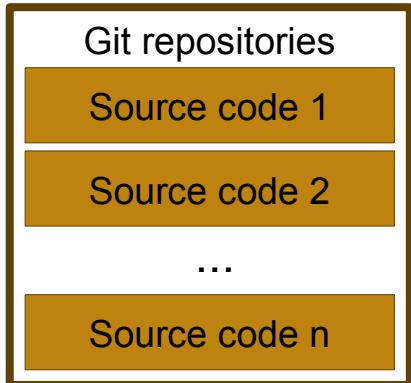
Docker image 2
+ Database E

Docker image 2
+ Database G



EPICS IOC's

- Hardware agnostic where possible
- Stateless (exception autosave)
- **Flexible: mix and match base+modules.**
- **Stable: a change to base or a module shouldn't randomly break IOC's.**



Containers deployed to production

Docker image
1:v0.1 +
Database A

Docker image
1:v0.2 +
Database D

Docker image
2:v0.1 +
Database E

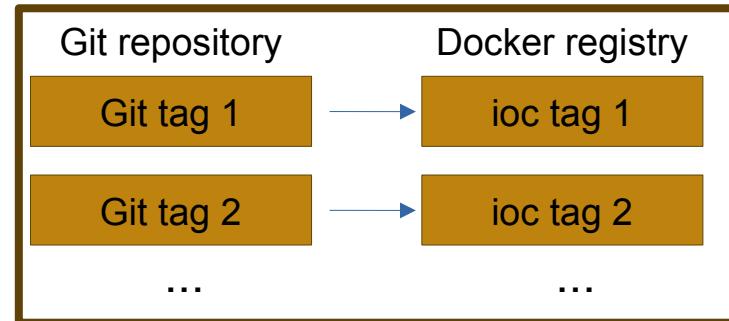
Docker image
2:v0.1 +
Database G



EPICS IOC's

- Hardware agnostic where possible
- Stateless (exception autosave)
- **Flexible: mix and match base+modules.**
- **Stable: a change to base or a module shouldn't randomly break IOC's.**

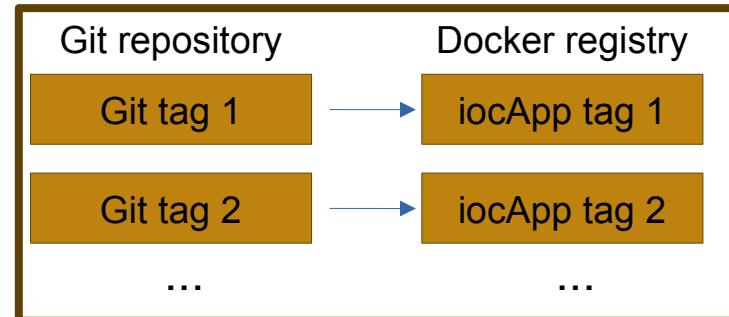
EPICS base + modules + iocApp + database



EPICS base + modules



EPICS base + modules + iocApp



EPICS IOC's

Developer A

EPICS base + modules

EPICS base + modules + iocApp

EPICS base + modules + iocApp
+ database

Developer B

EPICS base + modules

EPICS base + modules + iocApp

EPICS base + modules + iocApp
+ database

Production

EPICS base + modules

EPICS base + modules + iocApp

EPICS base + modules + iocApp
+ database



Gitlab CI/CD

Local git repository

Developer A

EPICS base + modules

EPICS base + modules + iocApp

EPICS base + modules + iocApp
+ database

Local git repository

Developer B

EPICS base + modules

EPICS base + modules + iocApp

EPICS base + modules + iocApp
+ database

Central gitlab server + CI/CD + docker registry

Production

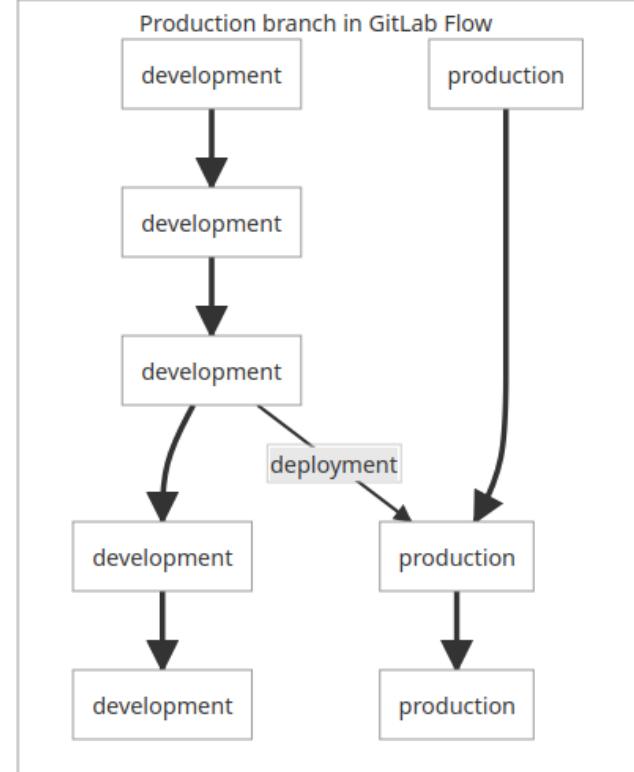
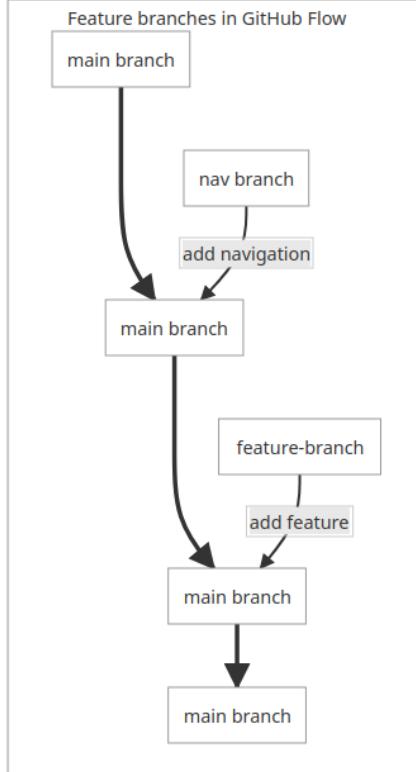
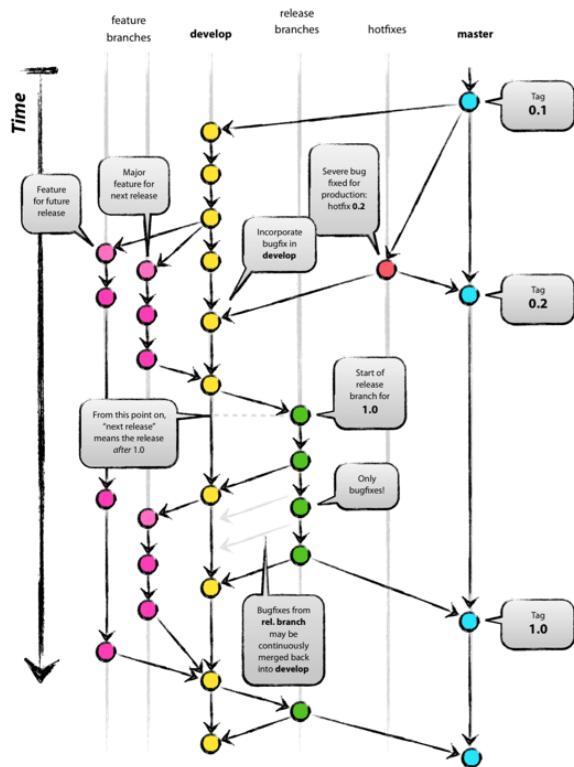
EPICS base + modules

EPICS base + modules + iocApp

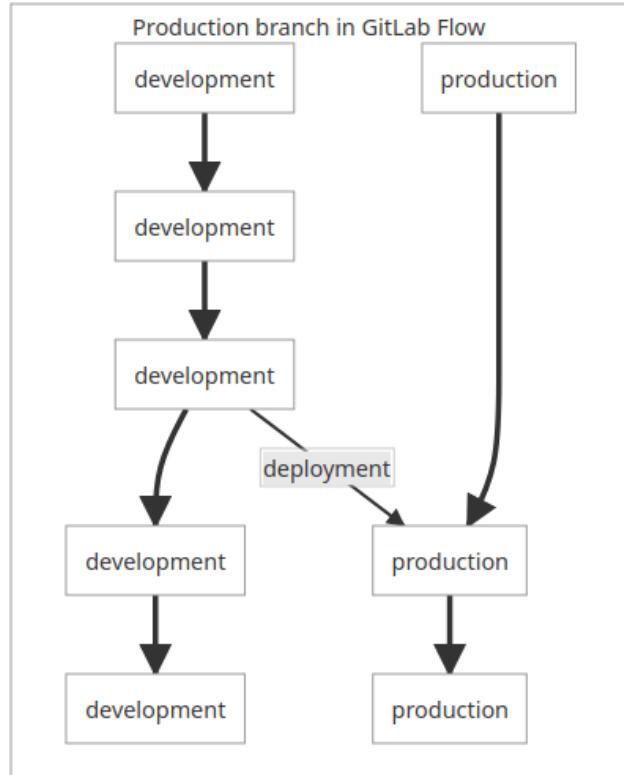
EPICS base + modules + iocApp
+ database



Gitlab CI/CD



Gitlab CI/CD



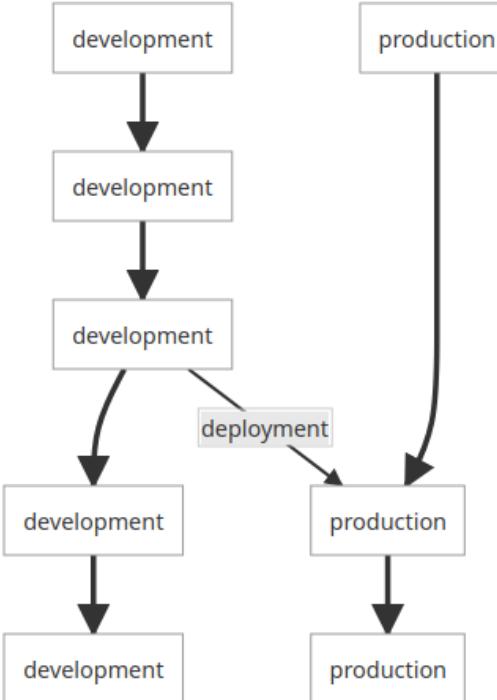
Gitlab
repository

Container
registry



Gitlab CI/CD

Production branch in GitLab Flow



Gitlab
repository

Container
registry

Pipeline Needs Jobs 3 Tests 0

Group jobs by Stage Job dependencies

Upstream

<
 Baseautobuild #1828
 Multi-project

build

<
 build-iocApp-image
 build-test-iocApp

test

<
 child-pipeline
 Trigger job

Downstream

<
 child-pipeline
 #1832
 Child

Upstream

<
 iocApp2Base
 #1832
 Multi-project

build

<
 build-ioc-image

test

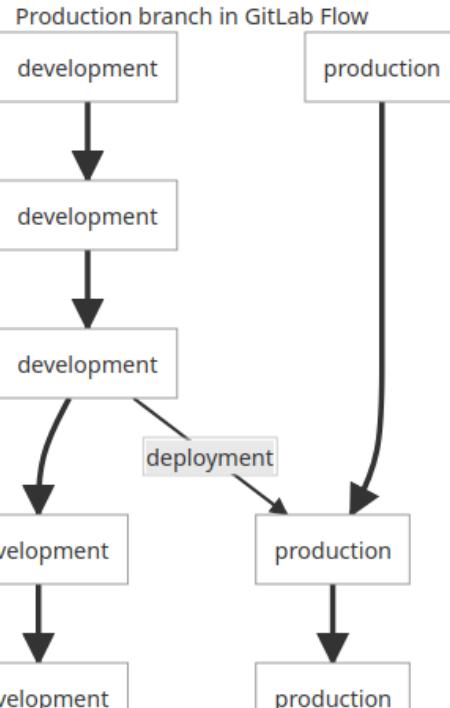
<
 test-ioc

deploy

start-live-test
stop-live-test



Gitlab CI/CD



Gitlab repository

Container registry

◀ upstream 2 Show all

> baseautobuild_ec401a8b29bee84f19d8f279b1ae5b278edf4abd Stop

▼ iocapp2base_4f8960381652d4345b938c86295d251e583ff19d production ▶ Stop

Success Latest Deployed #2 → dba207a8 Oct 16, 2022, 5:59 PM

demo ioc autobuild

Triggerer	Job	Branch	Tags
@christian	start-live-test	master	v0.0.1, v0.0.2

▼ production Stop

Success Latest Deployed #6 → 8d42df22 Oct 16, 2022, 6:31 PM

Autoupdate deploy

Triggerer	Job	Tags
@group_29_bot	start-ioc-production	v0.0.2

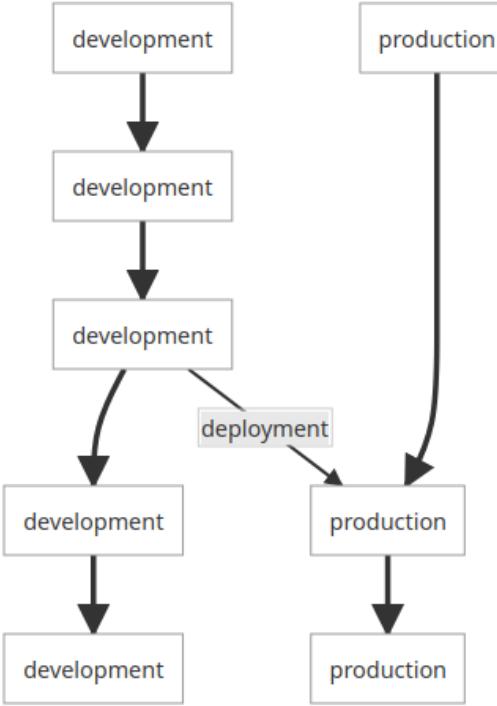
production

Status	ID	Triggerer	Commit	Job	Created	Deployed
success	#6		→ v0.0.2 → 8d42df22 Autoupdate deploy	start-ioc-prod...	Oct 16, 2022, ...	Oct 16, 2022, ...
success	#4		→ v0.0.1 → b727cbdb Autoupdate deploy	start-ioc-prod...	Oct 16, 2022, ...	Oct 16, 2022, ...



Gitlab CI/CD

Production branch in GitLab Flow



Gitlab
repository

Container
registry

Pipeline Needs Jobs 5 Tests 0

Group jobs by Stage Job dependencies

build

build-ioc-image

test

test-ioc

deploy

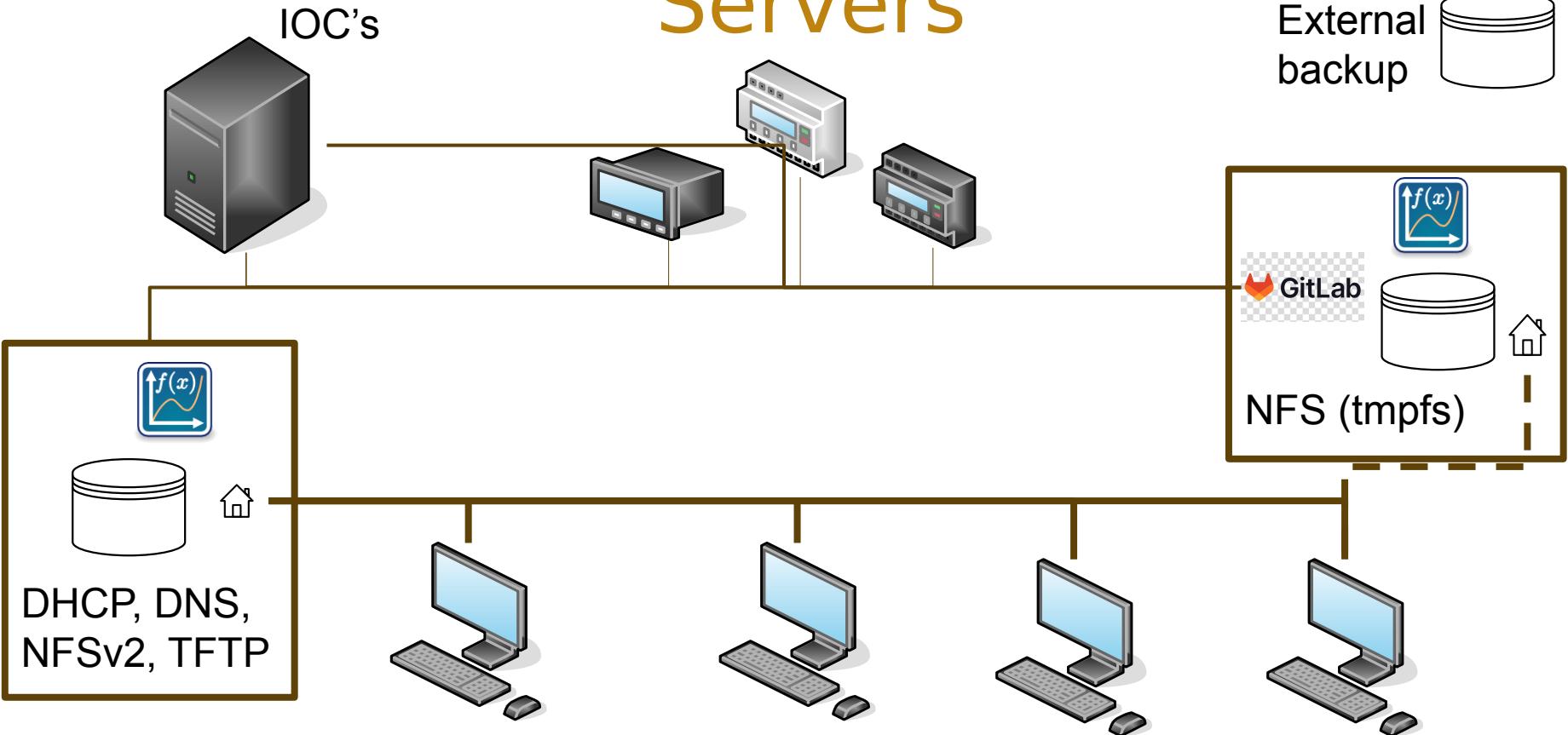
create-tag-and-upload-image

start-live-test

stop-live-test



Servers



Where do I get my ideas from?



<https://github.com/AustralianSynchrotron>



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



<https://github.com/lnls-sirius>

<https://github.com/pklaus>



<https://github.com/icshwi>



THANK YOU

