Test Harness

Continuous Testing on Hardware

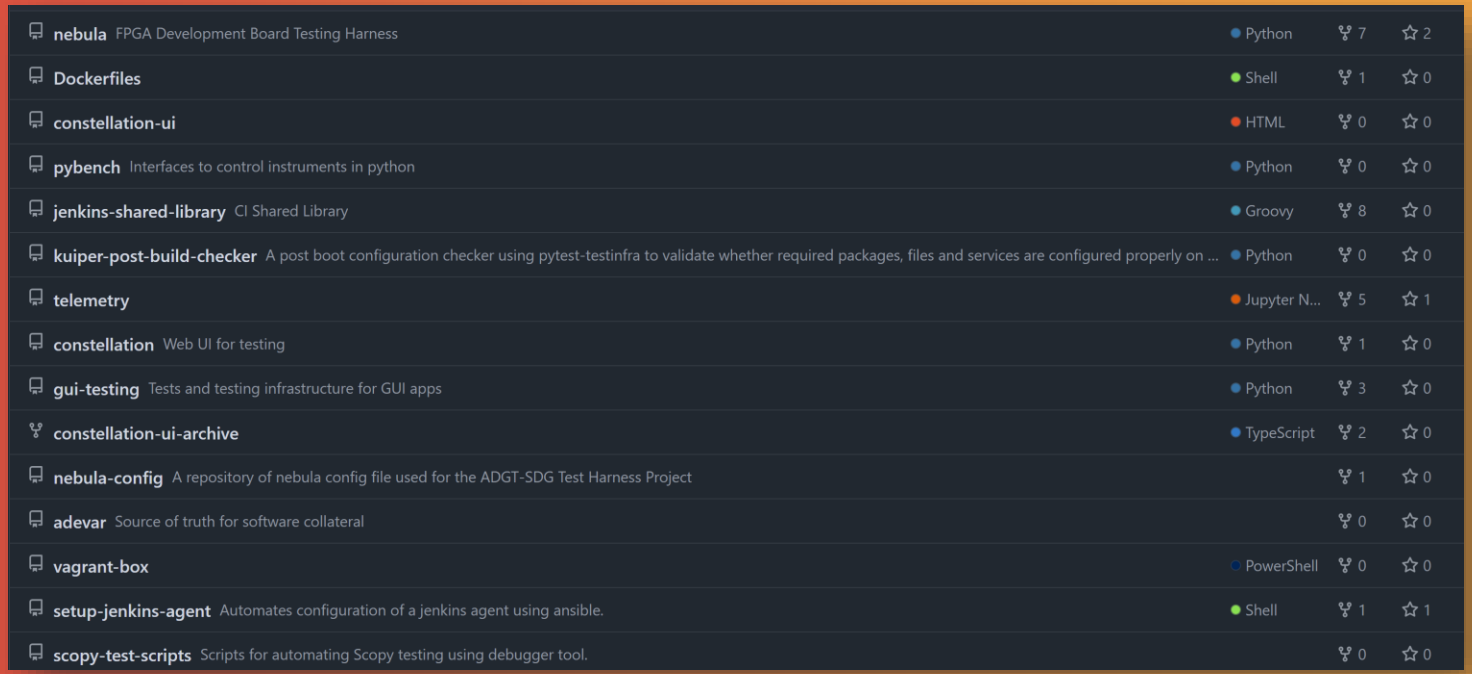
Agenda

- What it is
- Why create it
- How it’s made
- What challenges were encountered
- Where it’s currently used
- What’s left to do

whoami

- Software QA engineer working on embedded Linux systems
- Electronics engineering background
- Runner, not a fast one

DELL EMC


nebula FPGA Development Board Testing Harness Python 7 2
Dockerfiles Shell 1 0
constellation-ui HTML 0 0
pybench Interfaces to control instruments in python Python 0 0
jenkins-shared-library CI Shared Library Groovy 8 0
kuiper-post-build-checker A post boot configuration checker using pytest-testinfra to validate whether required packages, files and services are configured properly on ... Python 0 0
telemetry Jupyter N... 5 1
constellation Web UI for testing Python 1 0
gui-testing Tests and testing infrastructure for GUI apps Python 3 0
constellation-ui-archive TypeScript 2 0
nebula-config A repository of nebula config file used for the ADGT-SDG Test Harness Project 1 0
adevar Source of truth for software collateral 0 0
vagrant-box PowerShell 0 0
setup-jenkins-agent Automates configuration of a jenkins agent using ansible. Shell 1 1
scopy-test-scripts Scripts for automating Scopy testing using debugger tool. 0 0

sdgtt

What is this Test Harness? (aka board farm)

- Enables continuous testing across multiple hardware platforms
- Removes the burden from end developers about getting to a state with a specific board booted
- Logs, presents, and reports test results

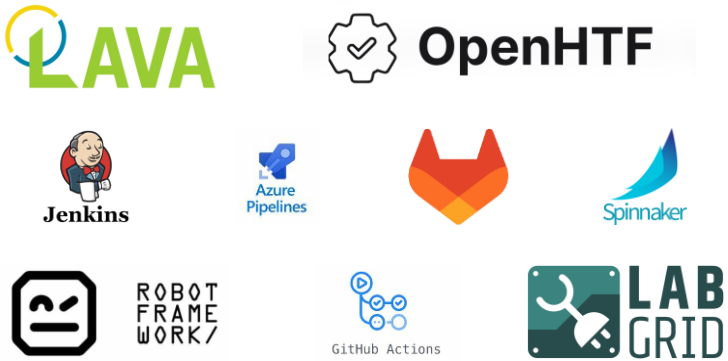
JIRA

What it’s should be

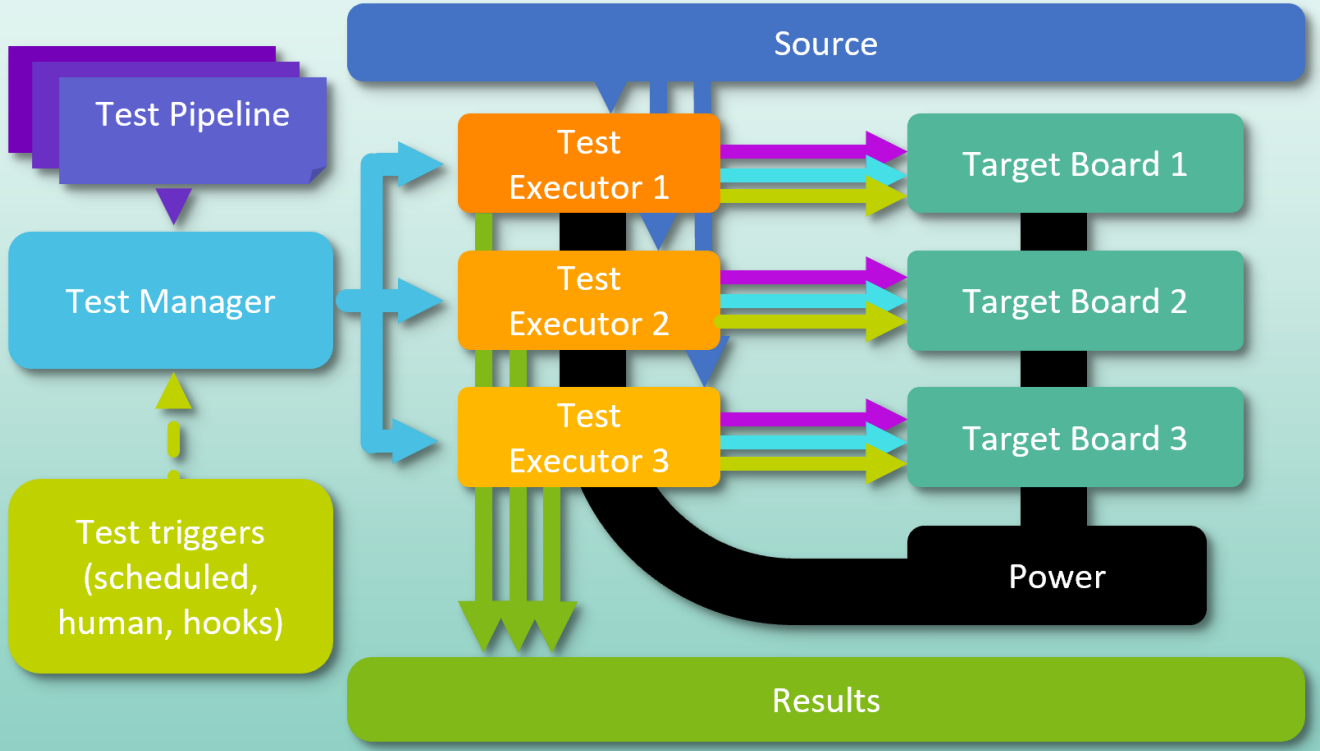
- Fully-automated
- System-level oriented
- Concurrent
- Flexible/Expandable
- Reliable

Why create it?

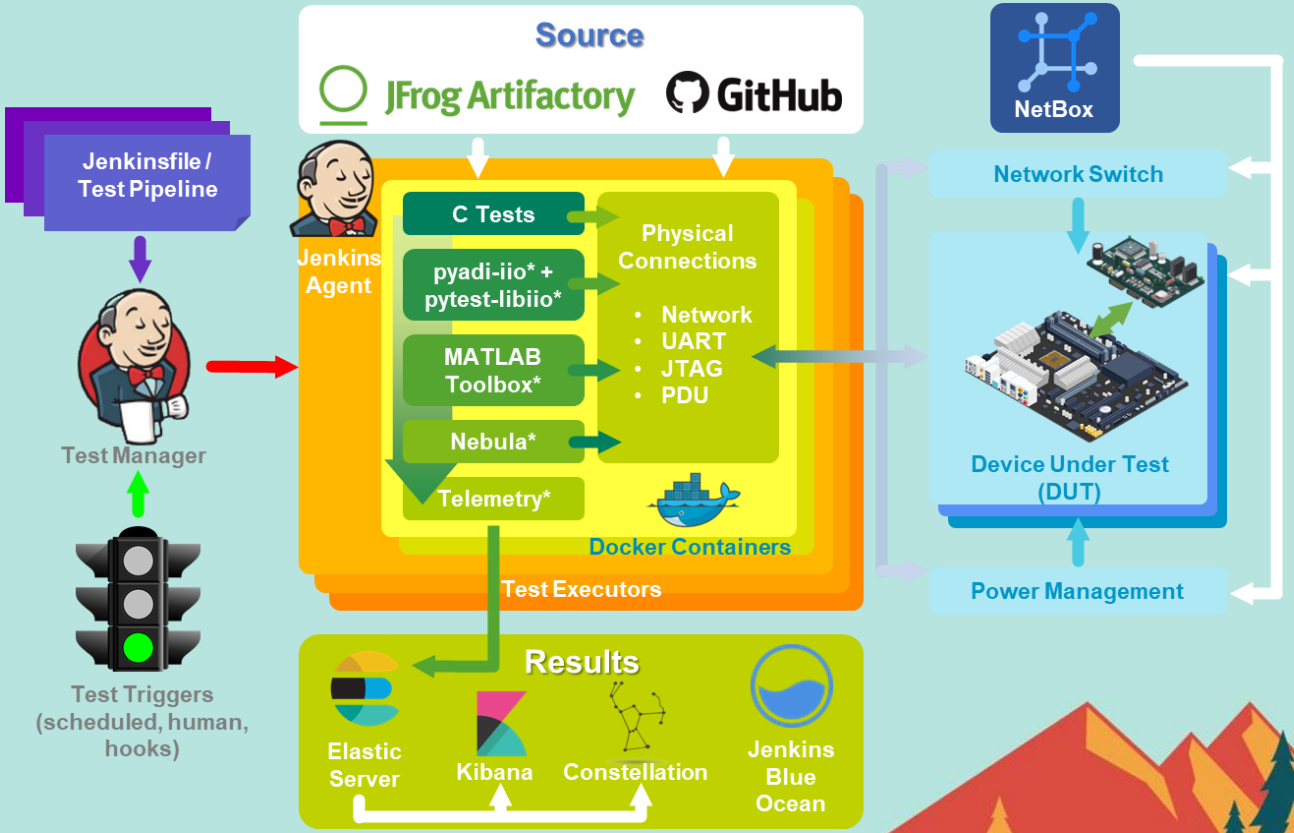
- Because existing frameworks don't quite fit
  - They are not hardware oriented
  - FPGA/SoC devices are not supported as a first-class test platform
  - Huge learning curves and maintenance burdens
- Researched solutions
  - Some of these frameworks are not fully ruled out



How it's made


Source
Test Pipeline
Test Manager
Test triggers (scheduled, human, hooks)
Test Executor 1
Test Executor 2
Test Executor 3
Target Board 1
Target Board 2
Target Board 3
Power
Results

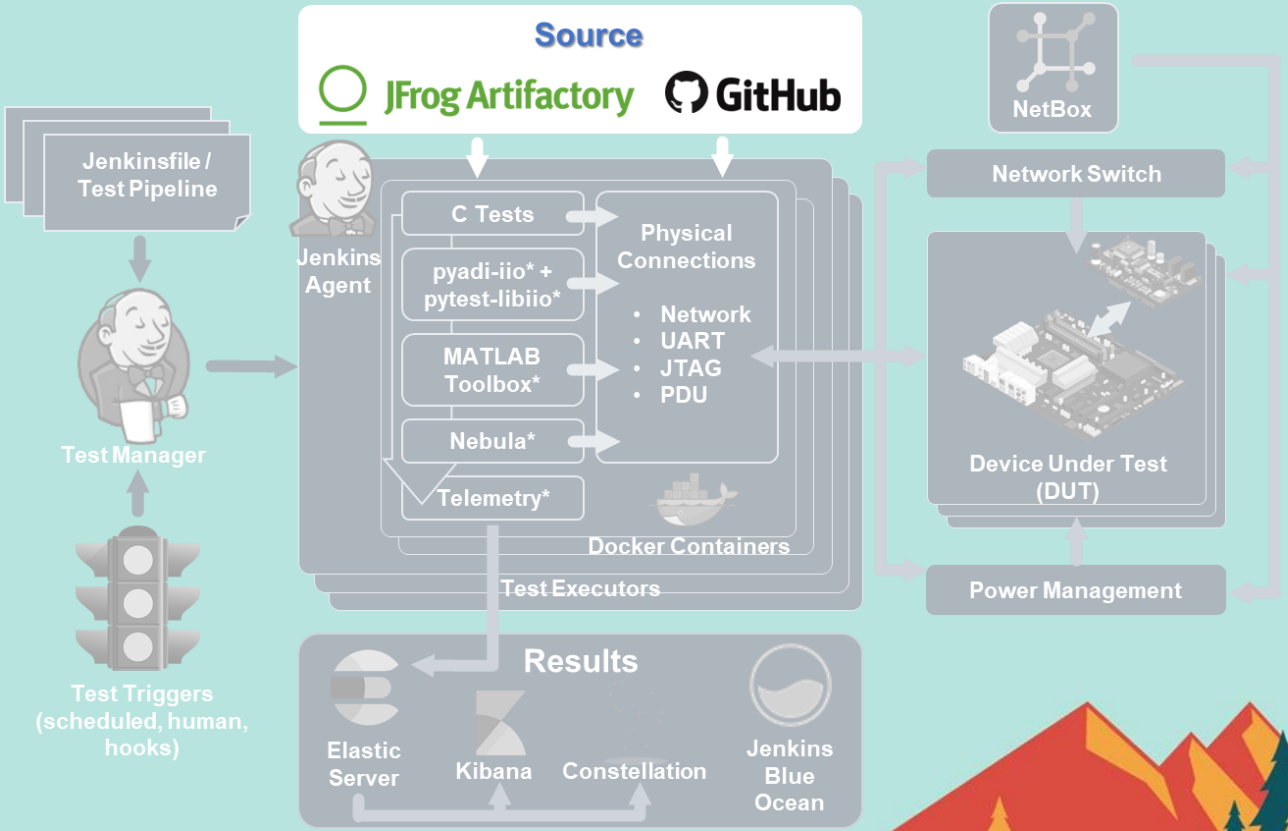
How it's made


Source
JFrog Artifactory
GitHub
NetBox
Jenkinsfile / Test Pipeline
Jenkins Agent
C Tests
pyadi-iio* + pytest-libiio*
MATLAB Toolbox*
Nebula*
Telemetry*
Physical Connections
Network
UART
JTAG
PDU
Docker Containers
Test Executors
Network Switch
Device Under Test (DUT)
Power Management
Test Manager
Test Triggers (scheduled, human, hooks)
Results
Elastic Server
Kibana
Constellation
Jenkins Blue Ocean

How it's made

1 Source Code and Binary Sources

- GitHub
- Artifactory Servers



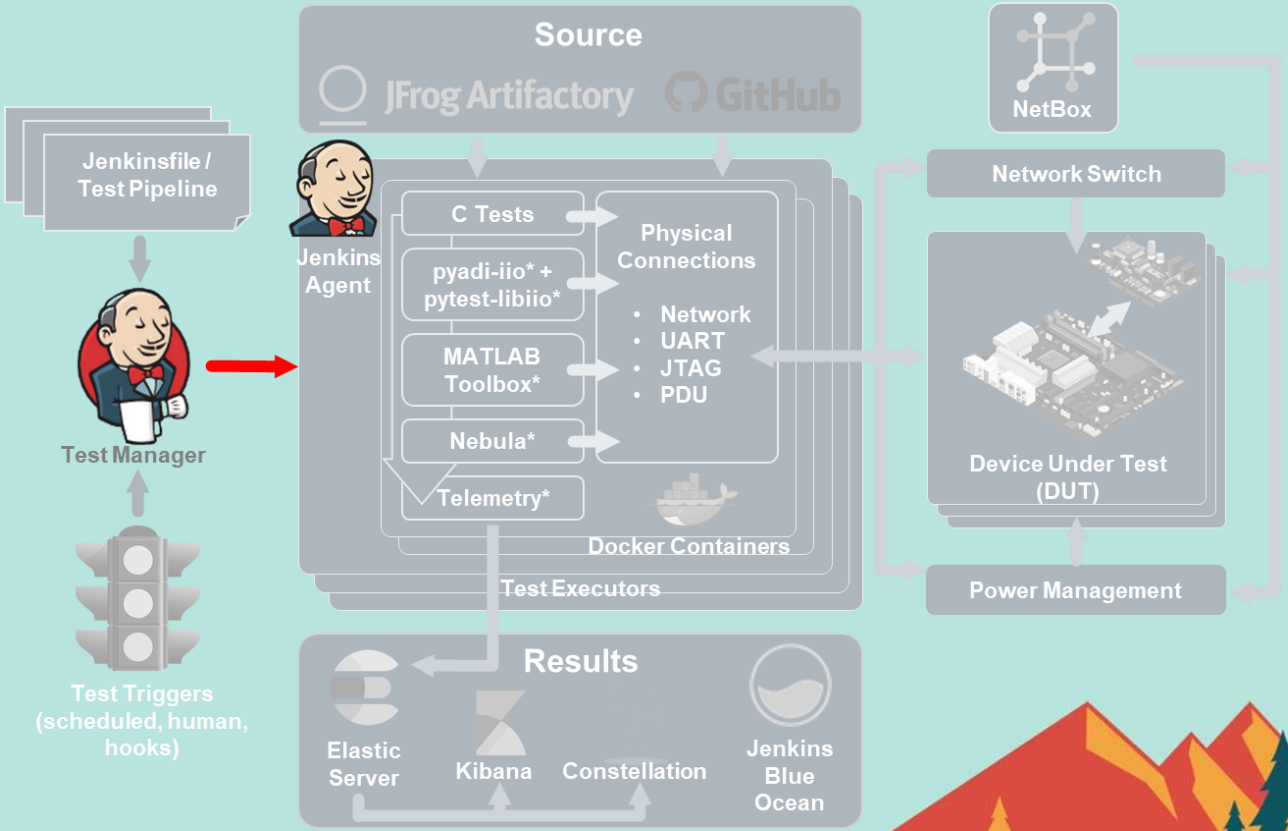
How it's made

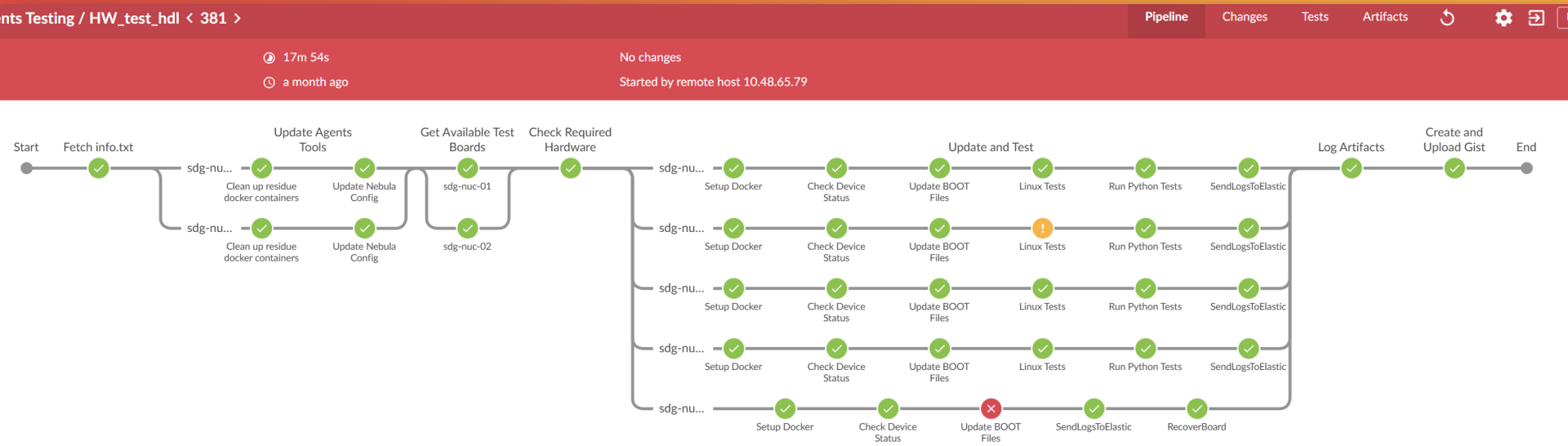
1 Source Code and Binary Sources

- GitHub
- Artifactory Servers

2 Test Manager and Agents

- Jenkins Server
- Jenkins Agent
- Jenkins Shared Library

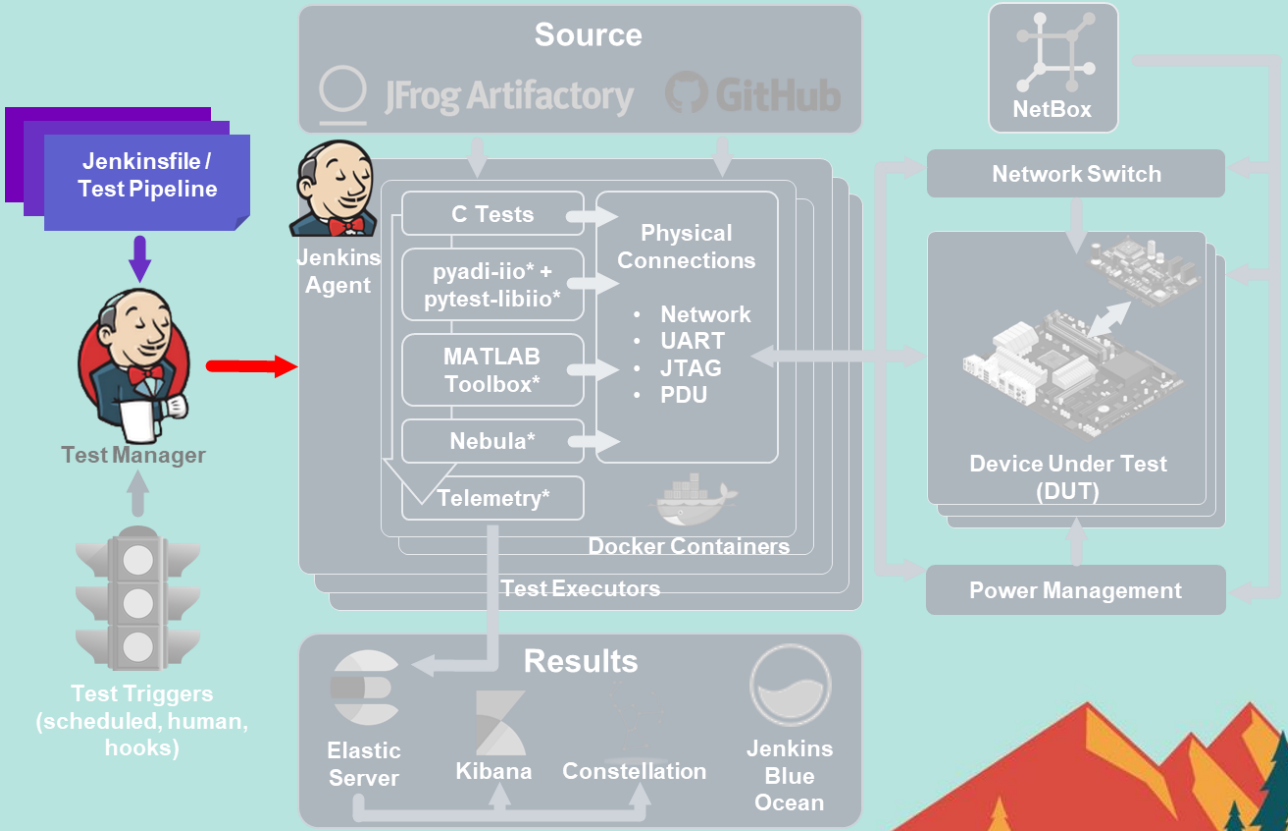



nts Testing / HW_test_hdl < 381 >
Pipeline
Changes
Tests
Artifacts
17m 54s
a month ago
No changes
Started by remote host 10.48.65.79
Start
Fetch info.txt
Update Agents Tools
sdg-nu...
Clean up residue docker containers
Update Nebula Config
sdg-nu...
Clean up residue docker containers
Update Nebula Config
Get Available Test Boards
sdg-nuc-01
sdg-nuc-02
Check Required Hardware
Update and Test
sdg-nu...
Setup Docker
Check Device Status
Update BOOT Files
Linux Tests
Run Python Tests
SendLogsToElastic
sdg-nu...
Setup Docker
Check Device Status
Update BOOT Files
Linux Tests
Run Python Tests
SendLogsToElastic
sdg-nu...
Setup Docker
Check Device Status
Update BOOT Files
Linux Tests
Run Python Tests
SendLogsToElastic
sdg-nu...
Setup Docker
Check Device Status
Update BOOT Files
Linux Tests
Run Python Tests
SendLogsToElastic
sdg-nu...
Setup Docker
Check Device Status
Update BOOT Files
SendLogsToElastic
RecoverBoard
Log Artifacts
Create and Upload Gist
End

How it's made

1 Source Code and Binary Sources
- GitHub
- Artifactory Servers

2 Test Manager and Agents
- Jenkins Server
- Jenkins Agent
- Jenkins Shared Library



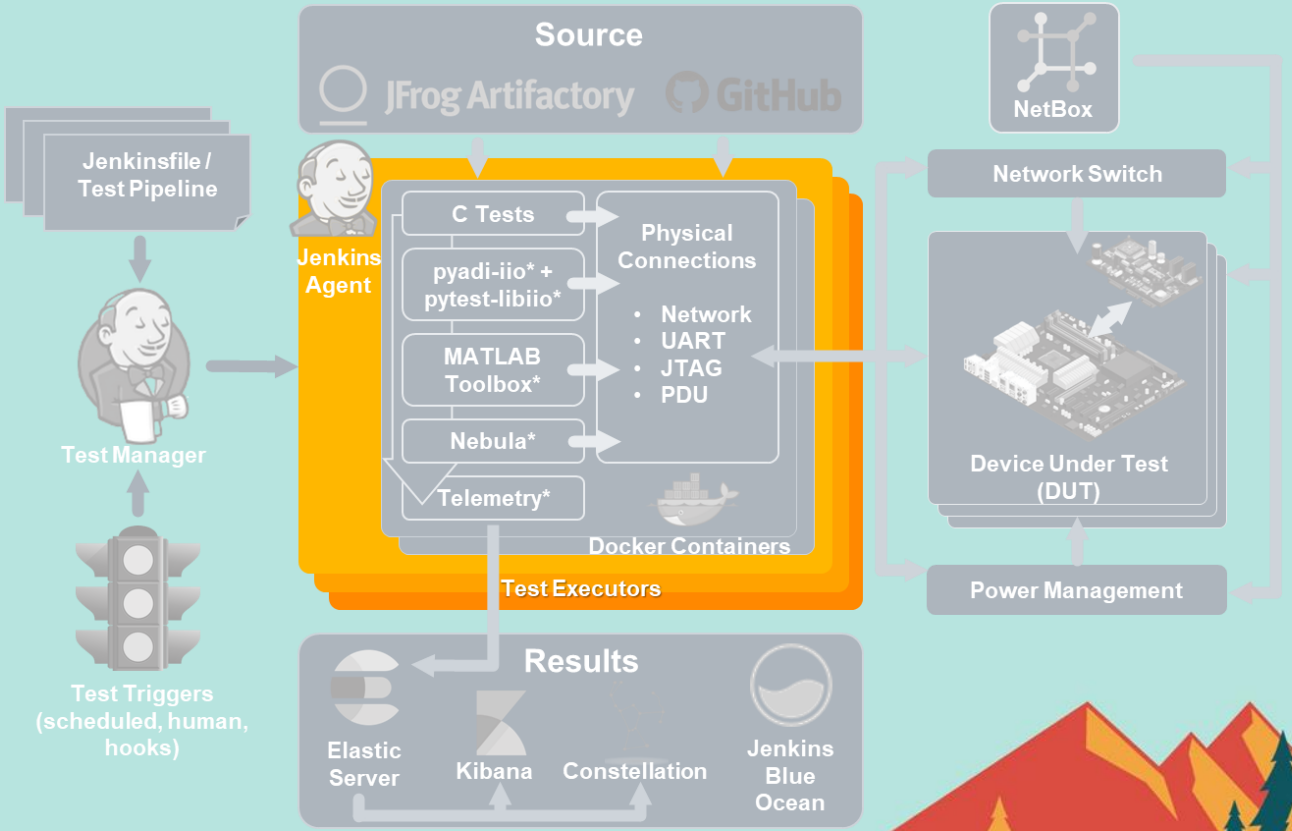
How it's made

1 Source Code and Binary Sources

- GitHub
- Artifactory Servers

2 Test Manager and Agents

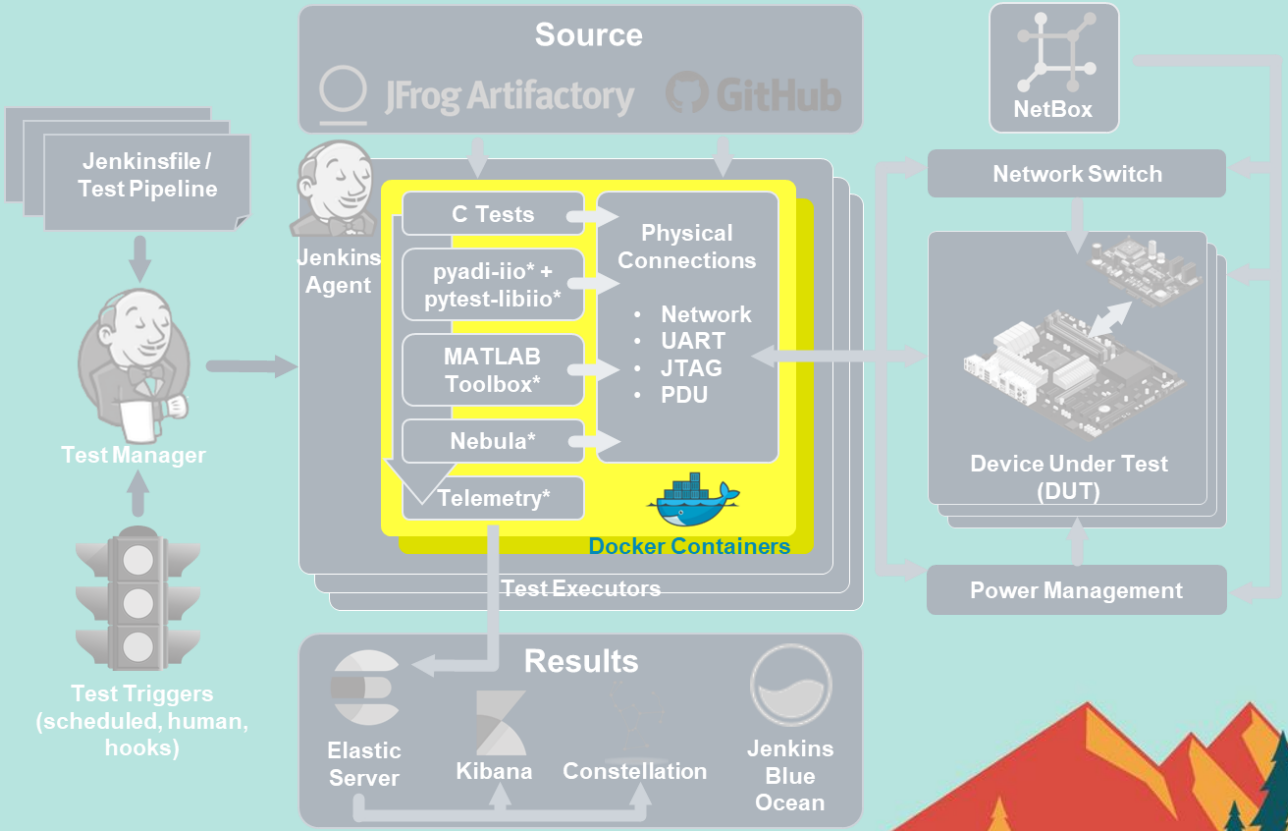
- Jenkins Server
- Jenkins Agent
- Jenkins Shared Library



How it's made

3 Environment Isolation

- Docker Containers



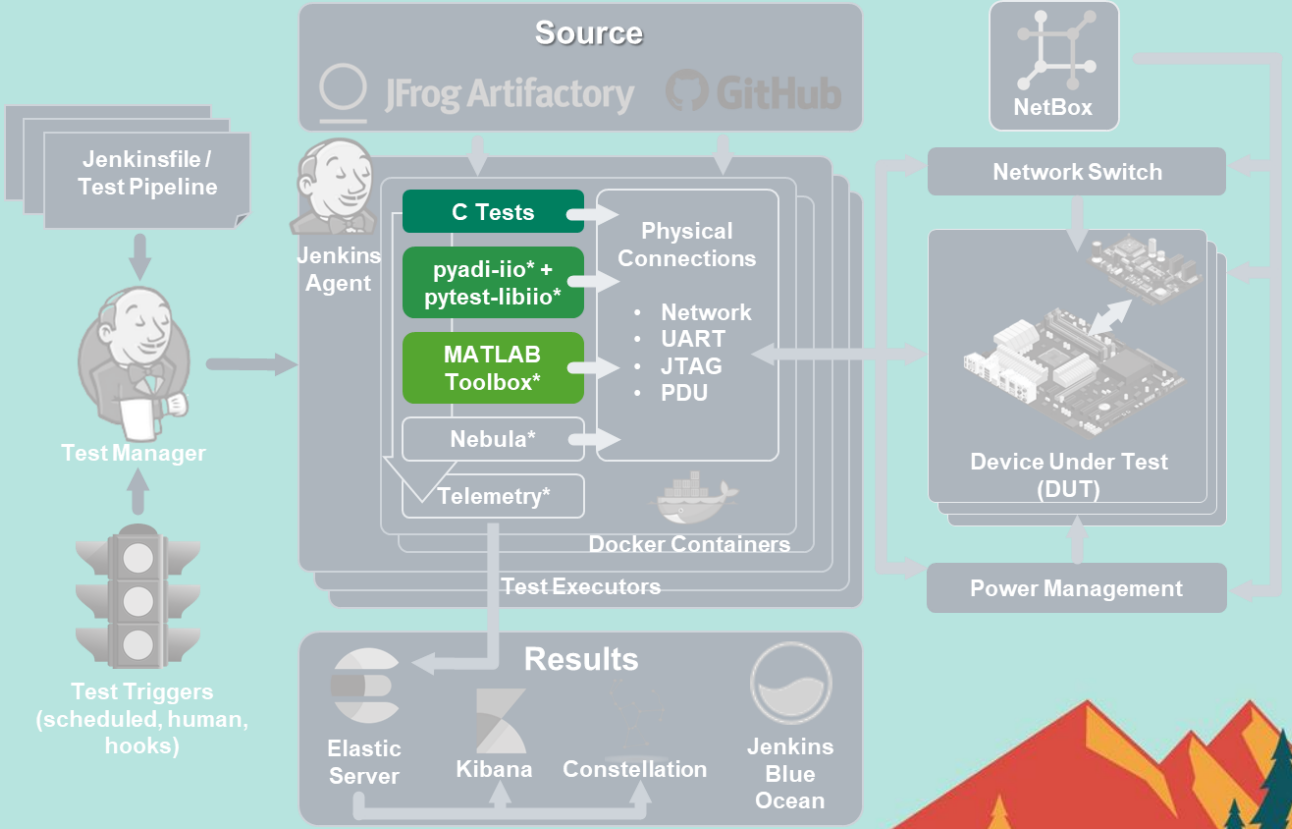
How it's made

3 Environment Isolation

- Docker Containers

4 Test Components

- PyADI-IIO tests*
- (Python-based tests)
- libad9361 tests*
- (C-based tests)
- Transceiver, High-speed Converter Toolboxes
- (MATLAB-based tests)



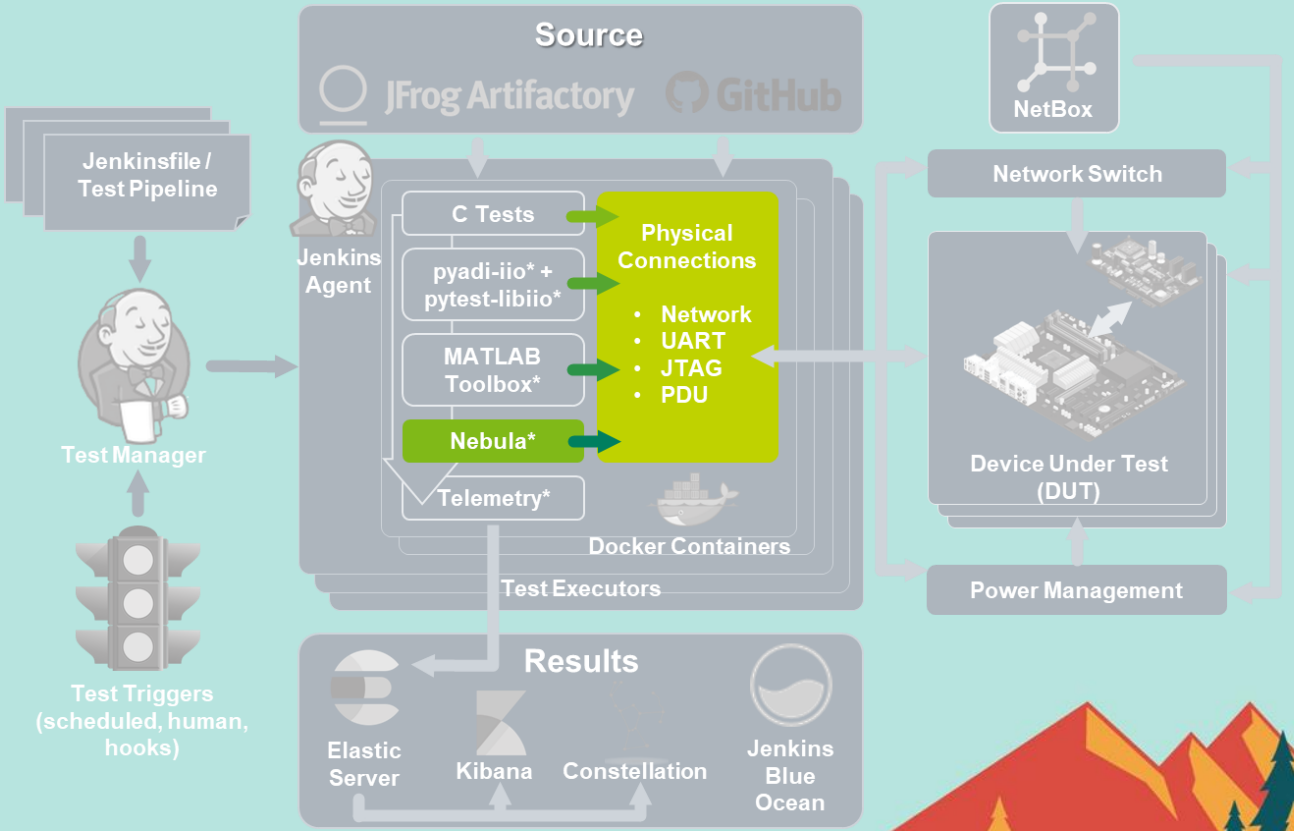
adpd1080.py	Update all years to 2025	5 months ago
adpd188.py	Update all years to 2025	5 months ago
adpd410x.py	Update all years to 2025	5 months ago
adrf5720.py	Update all years to 2025	5 months ago
adrv9002.py	Update all years to 2025	5 months ago
adrv9009.py	Add adrv9008-1/2 support	4 months ago
adrv9009_zu11eg.py	Update all years to 2025	5 months ago
adrv9009_zu11eg_fmcomms8.py	Update all years to 2025	5 months ago
adrv9009_zu11eg_multi.py	Update all years to 2025	5 months ago
adt7420.py	Update all years to 2025	5 months ago
adxl313.py	Fix merge	5 months ago
adxl345.py	Fix merge	5 months ago
adxl355.py	Fix merge	5 months ago
adxl380.py	Update all years to 2025	5 months ago

analogdevicesinc / pyadi-iio

How it's made

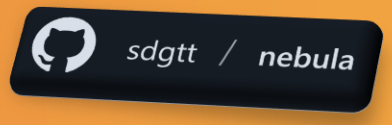
5 Board Interface Utility

- Nebula - Ethernet, UART, JTAG, USB, PDU, File Downloads



nebula show-log manager.update-boot-files --board-name=zynq-zed-adv7511-ad9361-fmcomms2-3 --folder=outs — Shell Script

```
+ nebula show-log manager.update-boot-files --board-name=zynq-zed-adv7511-ad9361-fmcomms2-3 --folder=outs
INFO | nebula.common : Depth of config: 1
INFO | nebula.common : board_name used: zynq-zed-adv7511-ad9361-fmcomms2-3
INFO | nebula.common : Depth of config: 1
INFO | nebula.common : board_name used: zynq-zed-adv7511-ad9361-fmcomms2-3
INFO | nebula.common : Depth of config: 1
INFO | nebula.common : board_name used: zynq-zed-adv7511-ad9361-fmcomms2-3
INFO | nebula.common : Depth of config: 1
INFO | nebula.common : board_name used: zynq-zed-adv7511-ad9361-fmcomms2-3
INFO | nebula.common : Depth of config: 1
INFO | nebula.common : board_name used: zynq-zed-adv7511-ad9361-fmcomms2-3
INFO | nebula.manager : SD-Card/microblaze based device selected
INFO | nebula.manager : Checking if Linux is accessible
INFO | nebula.uart : command response: Linux analog 4.14.0-gb9e7064 #441 SMP PREEMPT Fri Mar 29 08:45:14 GMT 2019 armv7l armv7l armv7l GNU/Linux
INFO | nebula.manager : Update board over SSH and reboot
INFO | nebula.network : Updating boot files over SSH
INFO | nebula.network : ssh command:ls /tmp/sdcard to root@192.168.10.2
WARNING | nebula.network : Exception raised: Encountered a bad command exit code!
```

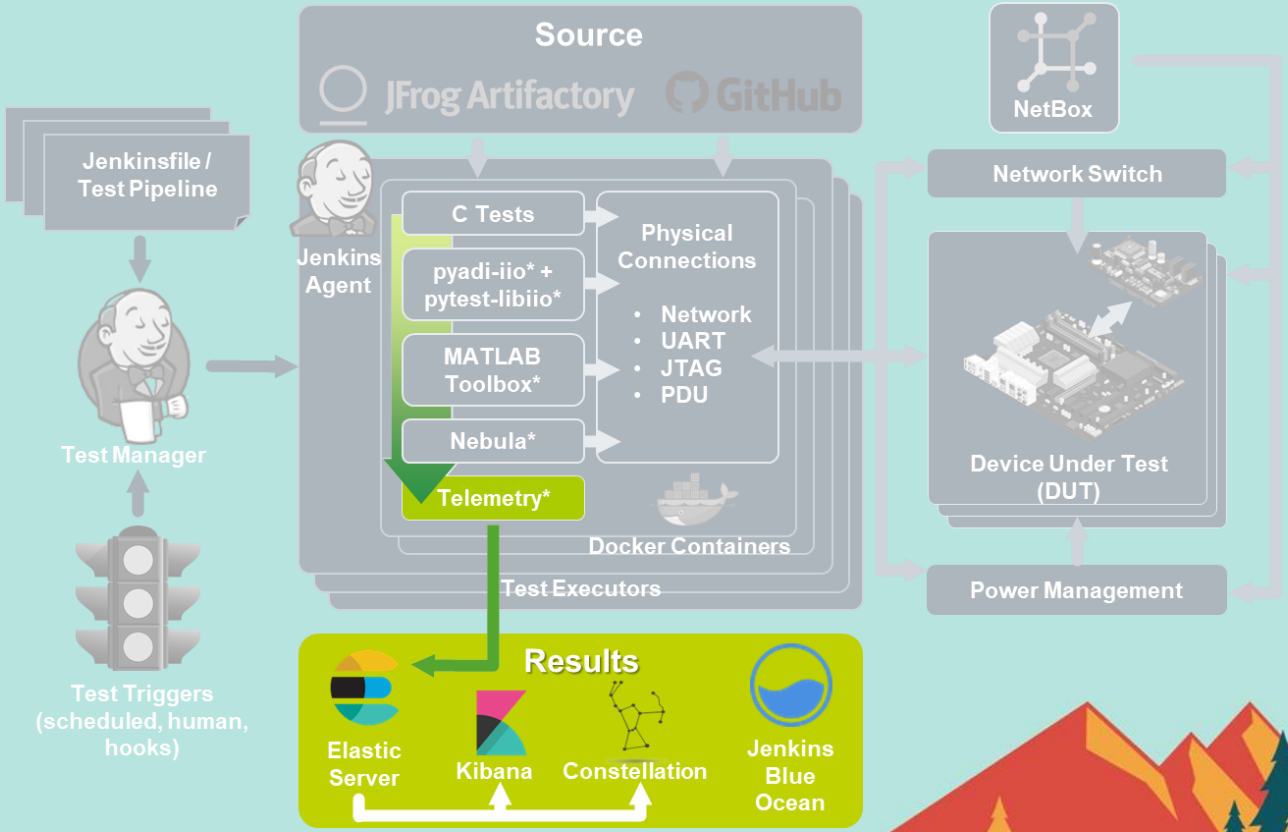
How it's made

5 Board Interface Utility

- Nebula - Ethernet, UART, JTAG, USB, PDU, File Downloads

6 Reporting

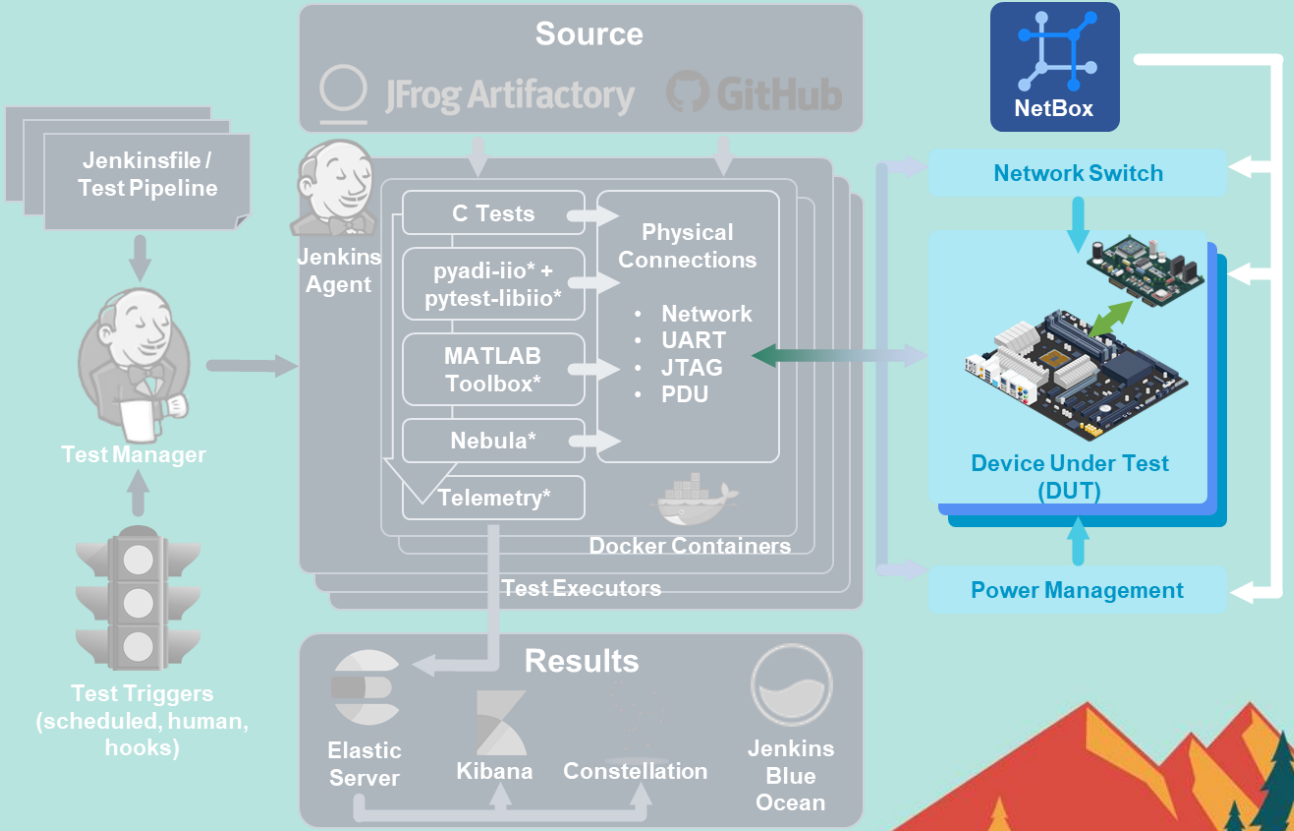
- Telemetry
- Elastic Server
- Kibana Dashboard
- Constellation
- Jenkins Blue Ocean

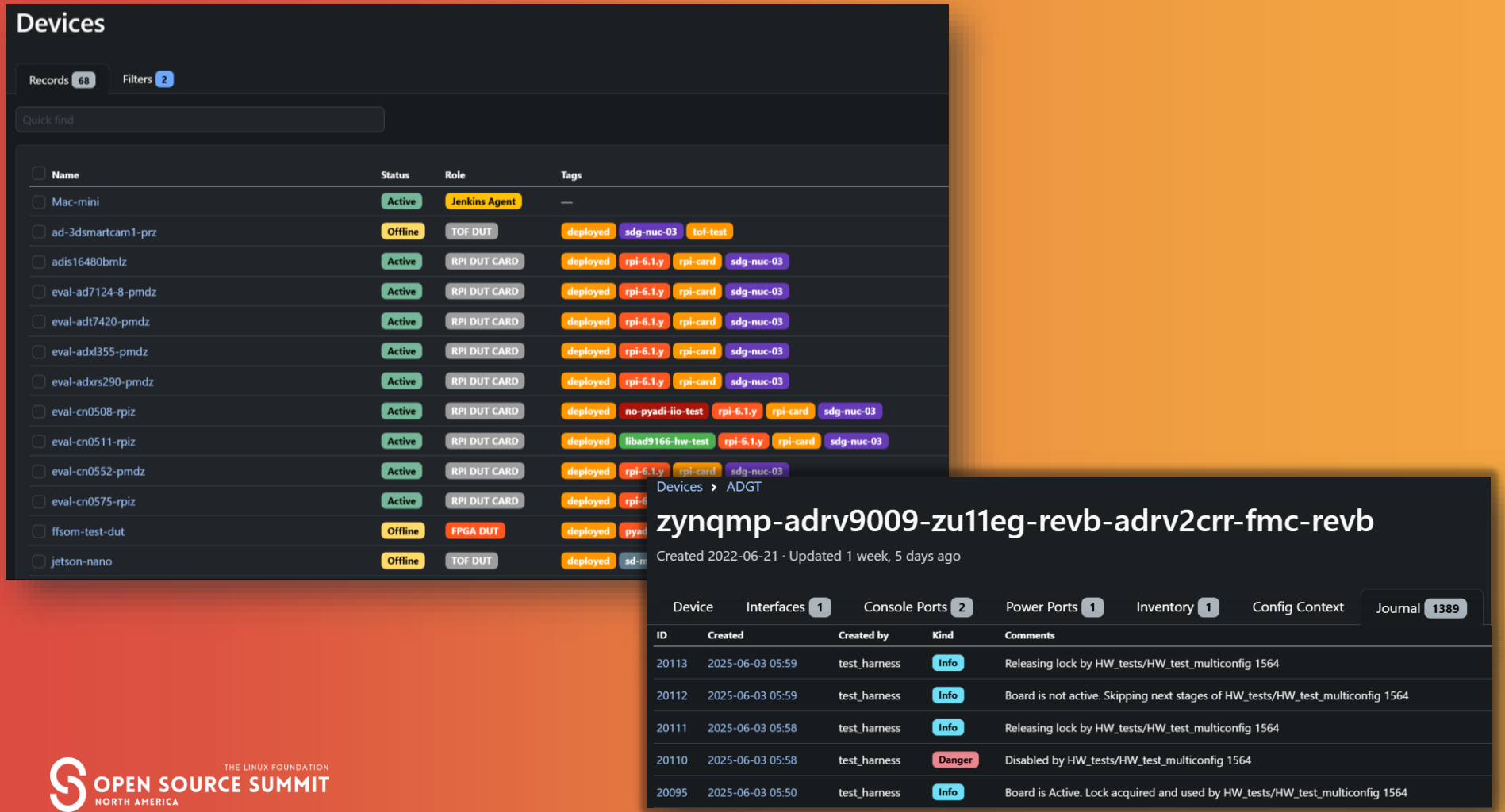


How it's made

Board Management

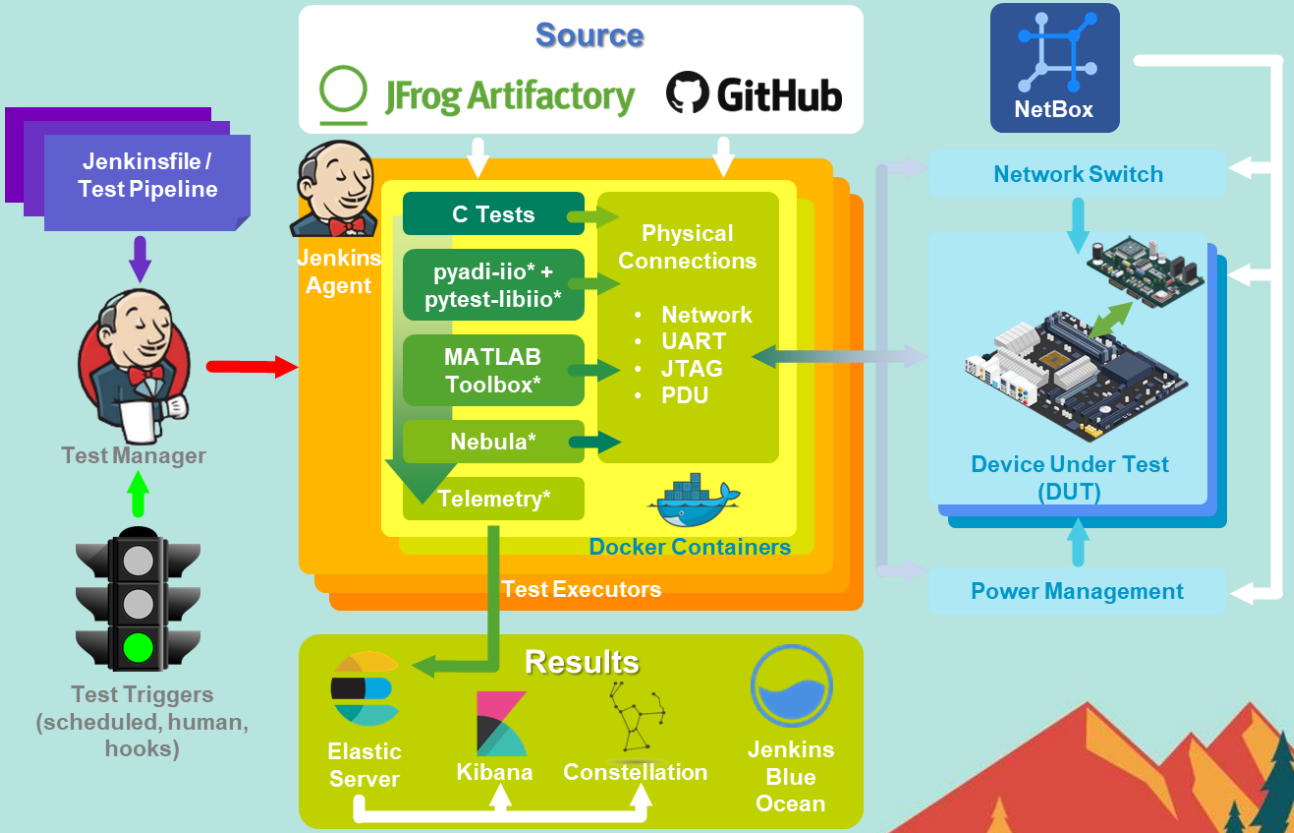
- NetBox
- nebula-config

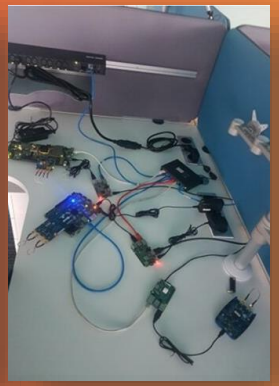


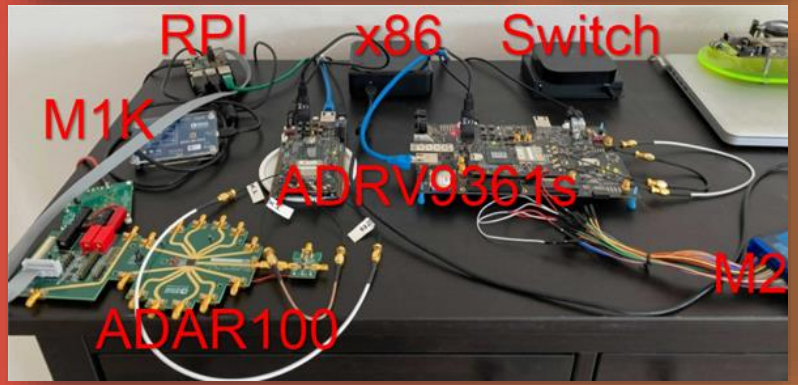


THE LINUX FOUNDATION
OPEN SOURCE SUMMIT
NORTH AMERICA

How it's made

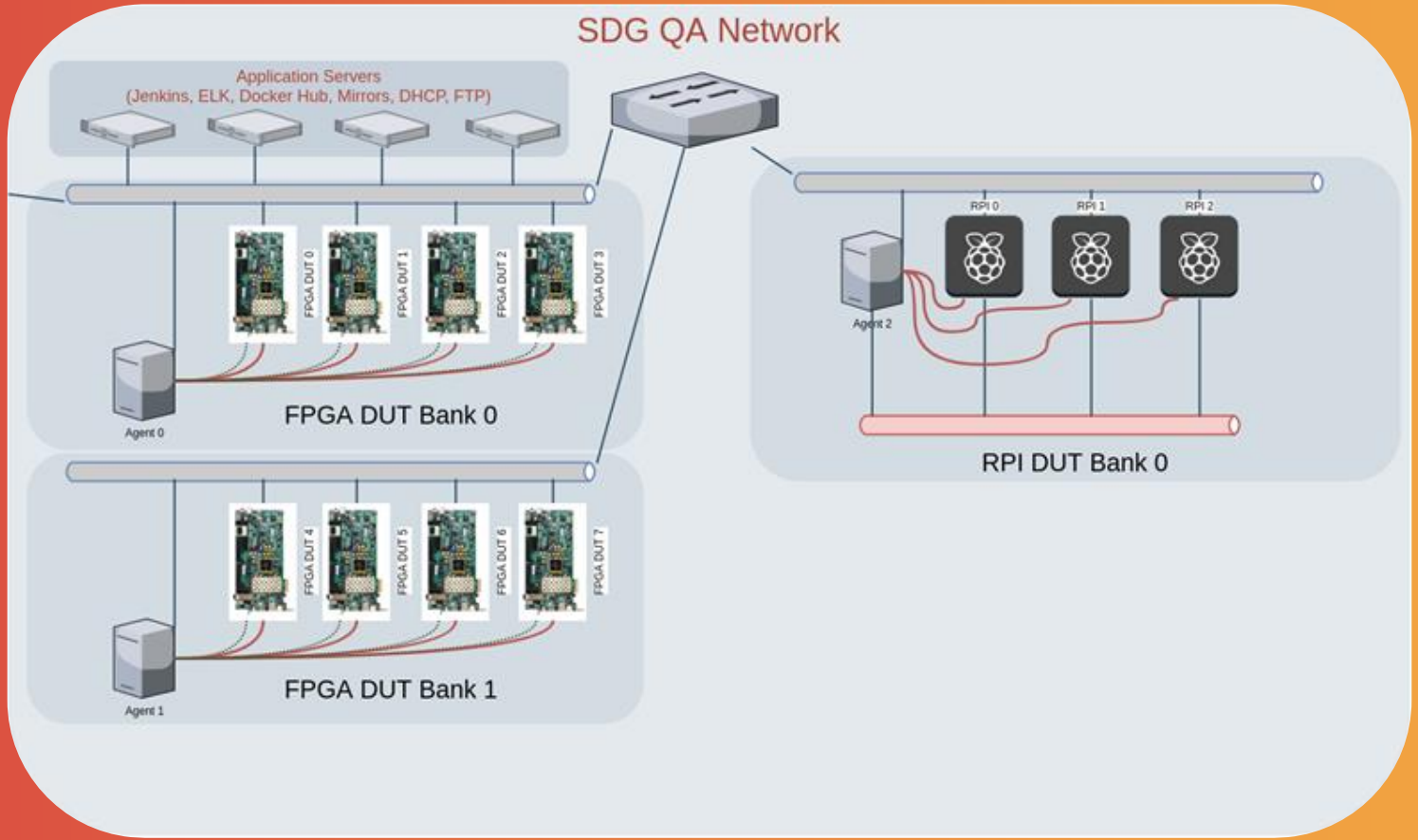

Source
JFrog Artifactory
GitHub
NetBox
Jenkinsfile / Test Pipeline
Jenkins Agent
C Tests
pyadi-iio* + pytest-libiio*
MATLAB Toolbox*
Nebula*
Telemetry*
Physical Connections
Network
UART
JTAG
PDU
Docker Containers
Test Executors
Network Switch
Device Under Test (DUT)
Power Management
Test Manager
Test Triggers (scheduled, human, hooks)
Results
Elastic Server
Kibana
Constellation
Jenkins Blue Ocean





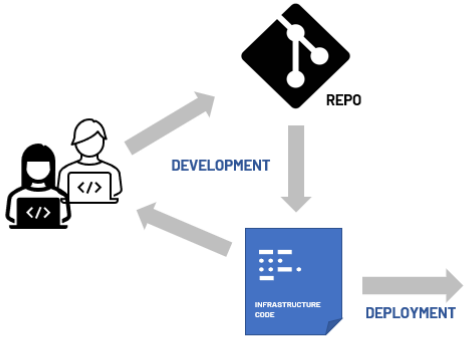
Current capacity

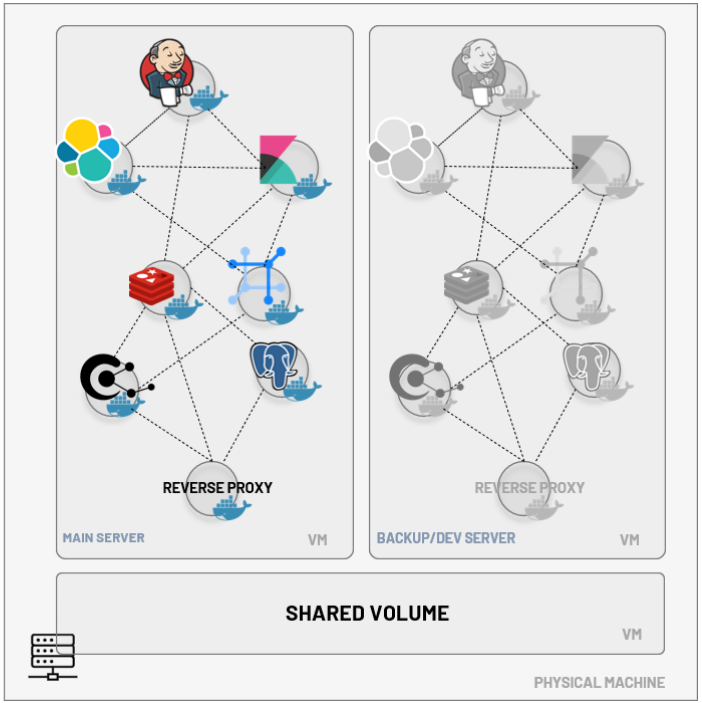
- 2 server racks
- 18 FPGA-based boards
- 9 RPi-based boards
- 4 bare-metal boards
- 6 agents
- 4 ADALM2000
- 1 ADALM-Pluto


SDG QA Network
Application Servers
(Jenkins, ELK, Docker Hub, Mirrors, DHCP, FTP)
FPGA DUT 0
FPGA DUT 1
FPGA DUT 2
FPGA DUT 3
Agent 0
FPGA DUT Bank 0
FPGA DUT 4
FPGA DUT 5
FPGA DUT 6
FPGA DUT 7
Agent 1
FPGA DUT Bank 1
RPI 0
RPI 1
RPI 2
Agent 2
RPI DUT Bank 0

Infrastructure Deployment

TEST HARNESS INFRASTRUCTURE AS A CODE WORKFLOW





- Infrastructure as Code
- Containerized services

Lifelines (when they work)

- USB-SD mux
- Plug-and-test support
- SD card etcher program

Where it's currently used

- Artifacts
  - Kuiper Linux boot files
  - RPI boot files
  - no-os binaries
  - plutosdr firmware
  - m2k firmware
- Repositories
  - ADI MATLAB Toolboxes
  - libad9166-iio
  - libad9361-iio
  - libm2k
  - pyadi-iio
  - HDL
  - Linux

What's left to do

- Stability efforts
  - Jenkins Shared Library Refactor
  - Stable board recovery stage on all platforms
- Make use of Kubernetes to manage the services
- More docs!

Thank you!