Taking open source software to production readiness

With ONF distributed devops

Suchitra Vemuri
ONF Director of QA

2020

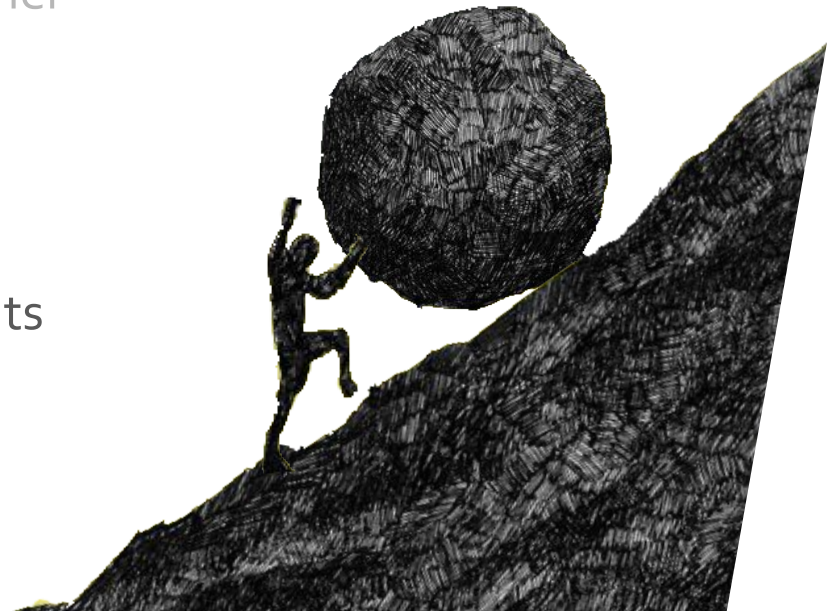
The Challenge

For a typical vendor to deliver a “carrier grade” production ready software to a service provider represents a big challenge….

The Challenge

For a typical vendor to deliver a "carrier grade" production ready software component to a service provider represents a big challenge

+ a solution with multiple components

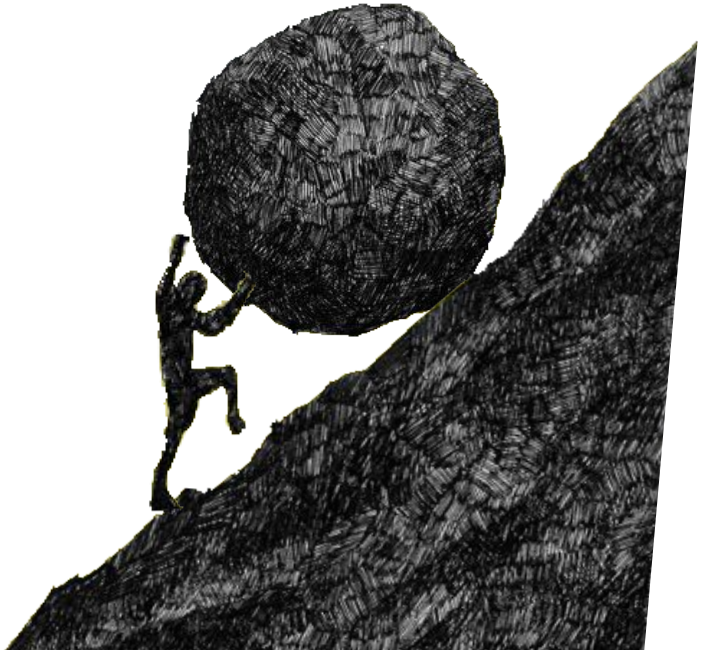


The Challenge

For a typical vendor to deliver a "carrier grade" production ready software component to a service provider represents a big challenge

+ a solution with multiple components

+ delivering with the new CI/CD paradigm



The Challenge

For a typical vendor to deliver a "carrier grade" production ready software component to a service provider represents a big challenge

+ a solution with multiple components

+ delivering with the new CI/CD paradigm

+ delivering it in an open source community with many companies participating with different levels of commitment makes it impossible

Operator challenge: bringing open source to production quality

How does one ensure quality software in an open source based solution?

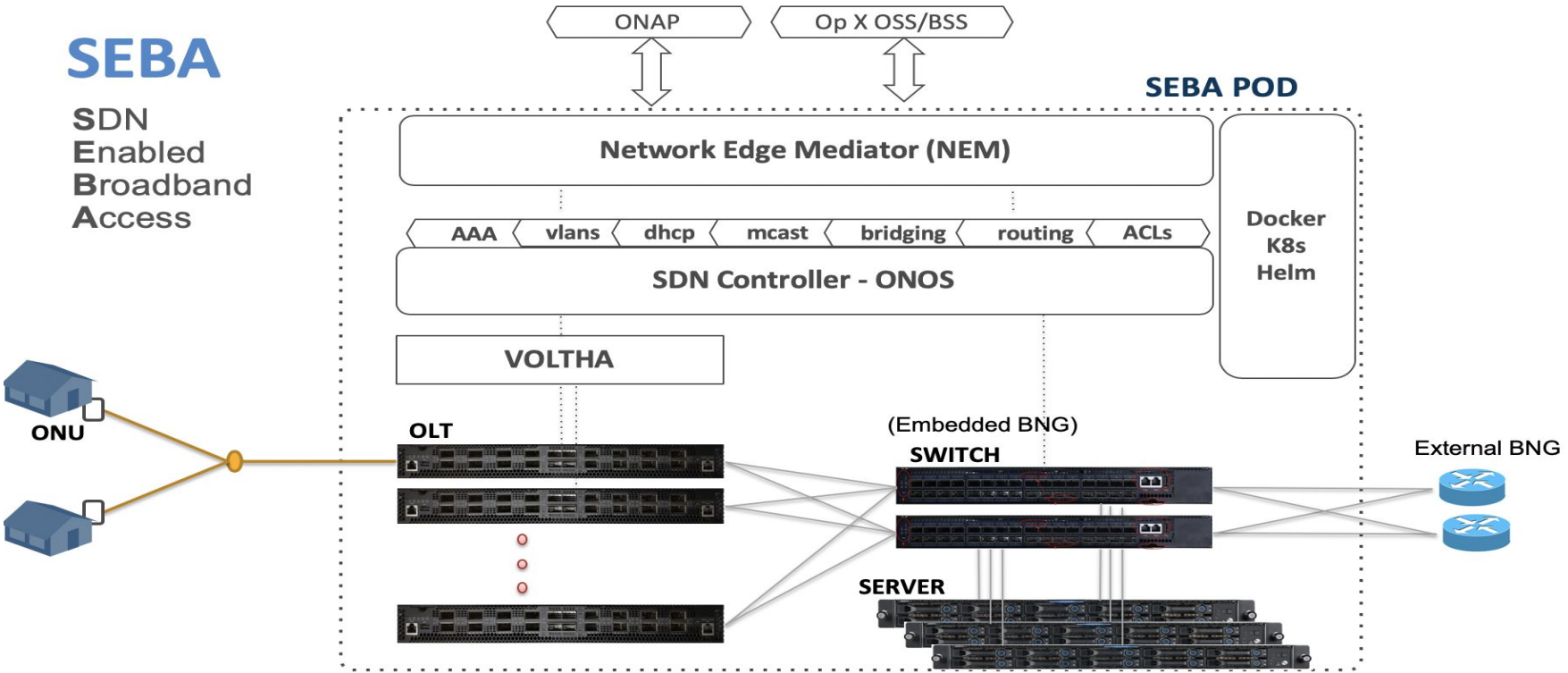
- How to organize a community spanning multiple companies with different processes?
- How to have ownership of issues? How to quickly resolve issues?
- How to establish common code/testing principles/guidelines?
- How to create common automation framework required to tame complexity?
- How to include real world production traffic which is required to get to production level?

Production releases achieved by ONF Community

- Comcast (Trellis, ONOS)
  - Trellis is a multi-purpose leaf-spine fabric running on ONOS and supporting distributed access networks and edge cloud applications.
  - Deployed in 20+ geographies with total 300k+ subscribers
- T-Mobile Poland (OMEC)
  - OMEC is the first full-featured, scalable, high performance open source EPC - providing fixed mobile (FMS) substitution in the Polish market.
  - First roll out - limited customers to run for 30-day grace period
- ATT, DT, TT (SEBA/VOLTHA - in progress)
  - SEBA supports a variety of virtualized access technologies at the edge of the carrier network. A key platform is VOLTHA used to virtualize access to PON infrastructure.

Taking SEBA/VOLTHA to production readiness …

Our Focus - SEBA/Voltha Project



Testing to production level

Comparing incumbent vendor vs open source

Vendor	Open Source
Single business unit (clear authority and responsibility)	Multiple organizations (authority and responsibilities need to be coordinated by ONF)
Multiple groups within the organization (development, multiple test groups)	Groups spread across organizations
Single process/workflow defined by the organization	Multiple processes/workflows
Tools/frameworks are well known	Tools/frameworks are different
Various types of testing (system, scale, performance, stress, soak)	Various types of testing (spread across community coordinated by ONF)
Vendor works closely with the operator during field trials/deployments	Closely work with the operator under ONF's coordination (with multiple vendors)
Multiple-year release cycles	Continuous release cycles

Can we achieve a better result in an open source environment?

Better test coverage

Faster issue discovery and resolution

Faster time to scale, performance, interoperability

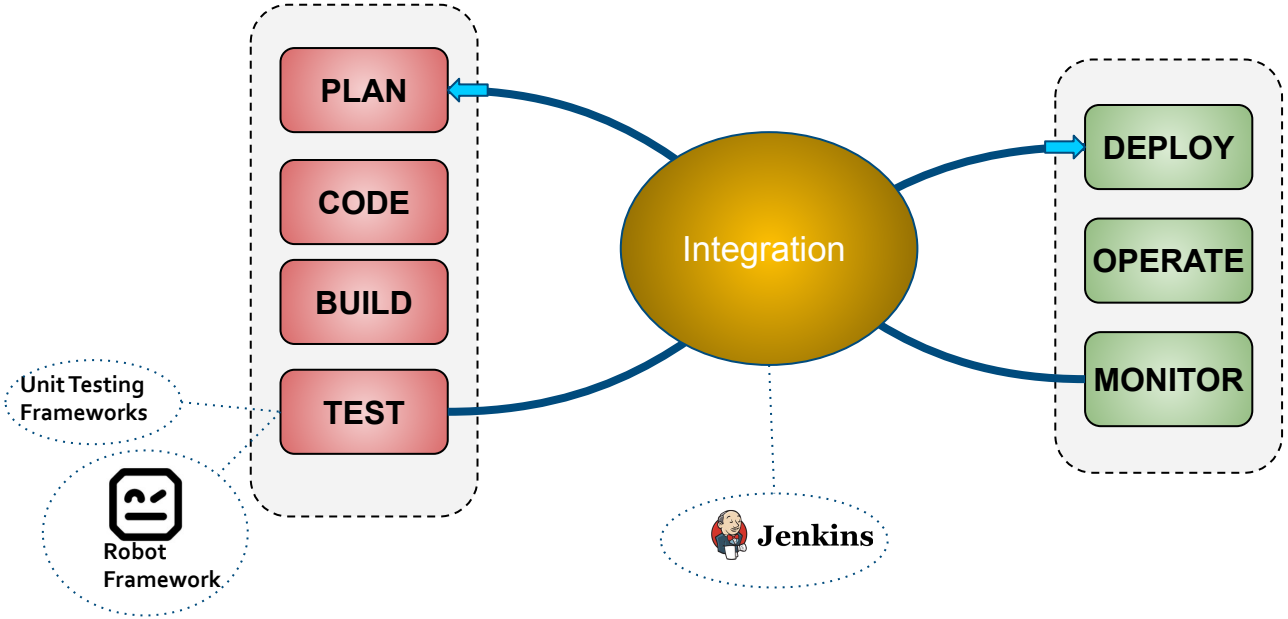
Faster cycles of incorporating changes

Yes - ONF community test process

ONF Community Testing - Best Practices

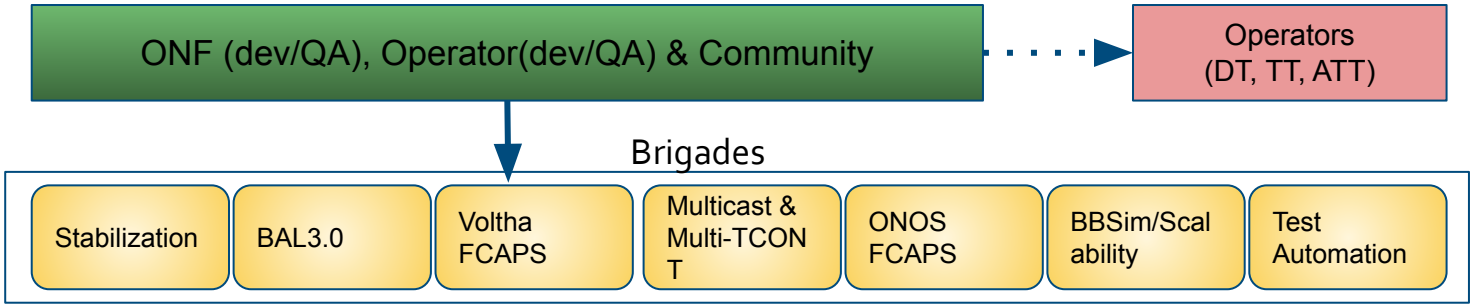
- Distributed DevOps
- Need to define good processes for CI/CD and establish coding/testing guidelines
- Establish a good continuous feedback model
- Establish good turn around times for handling issues
- Automate as much as possible with good tools/frameworks in place
- Automation required in
  - unit, functional, integration, scale, performance, soak/stability testing

DevOps Model



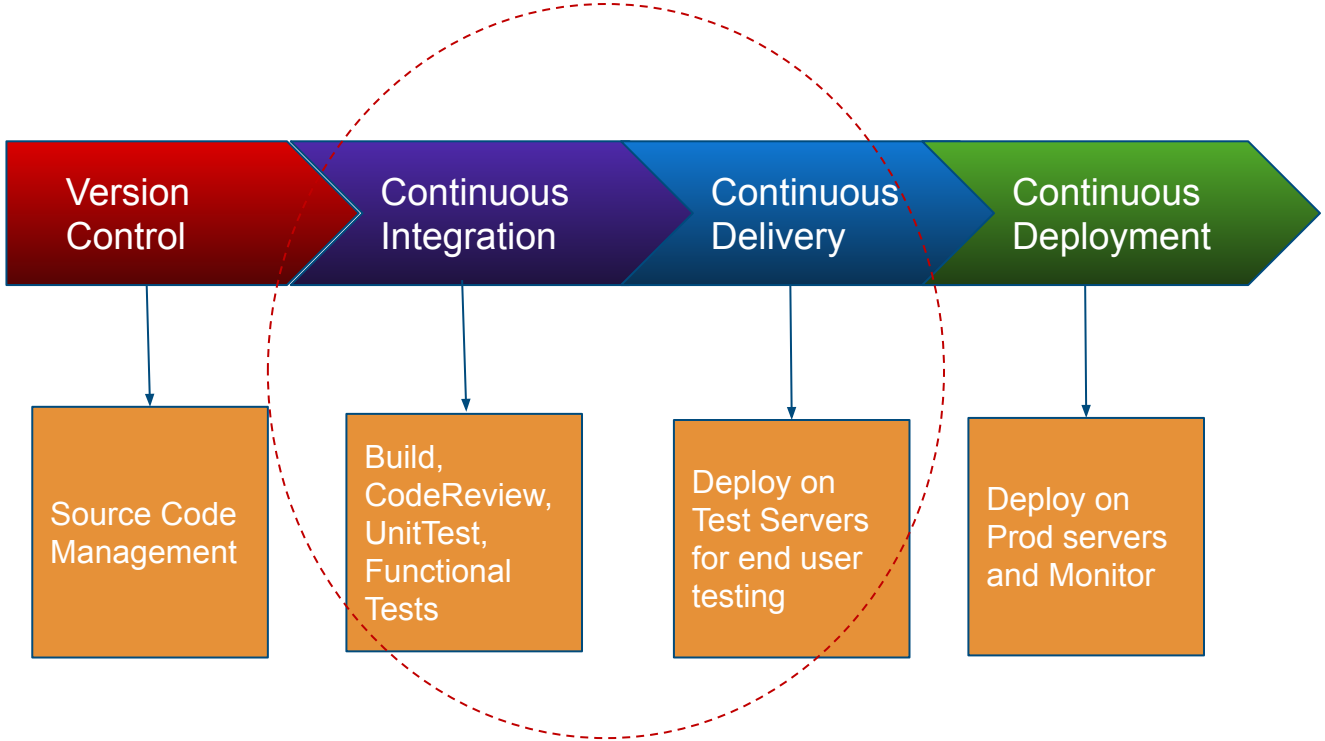
DevOps software strategy bridges gap between the development and operations side of the company and helps us to deliver quality software in time

ONF Distributed DevOps

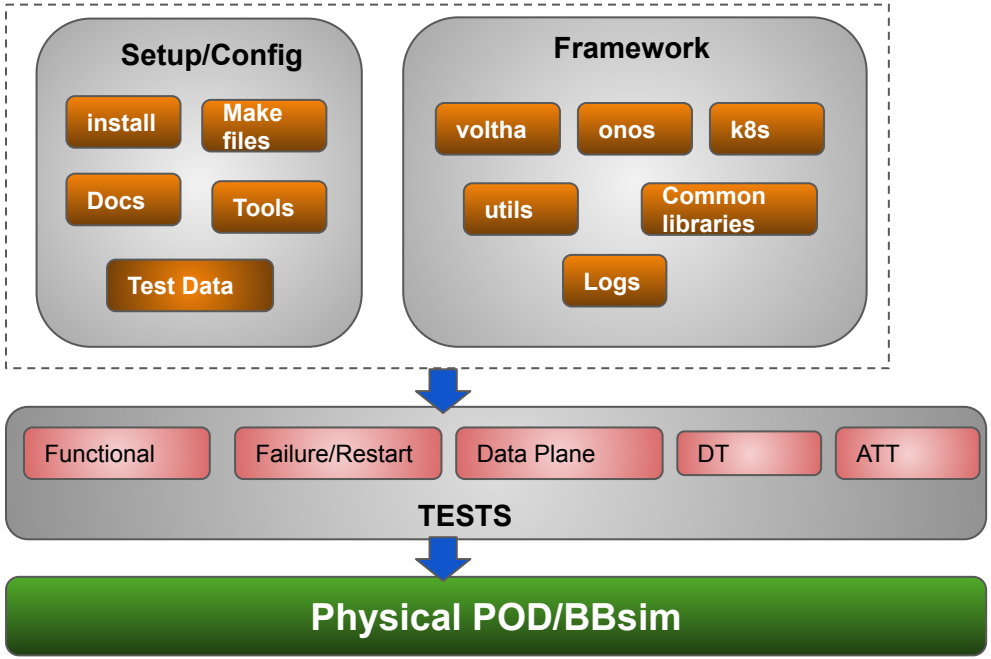


- Total of 80 contributors
- Team divided into brigades/work groups
- Process/workflows were defined
- Each brigade has specific goals and objectives
- Brigade progress was monitored/reviewed and shared with operators
- Communication established within and across groups

Testing in DevOps



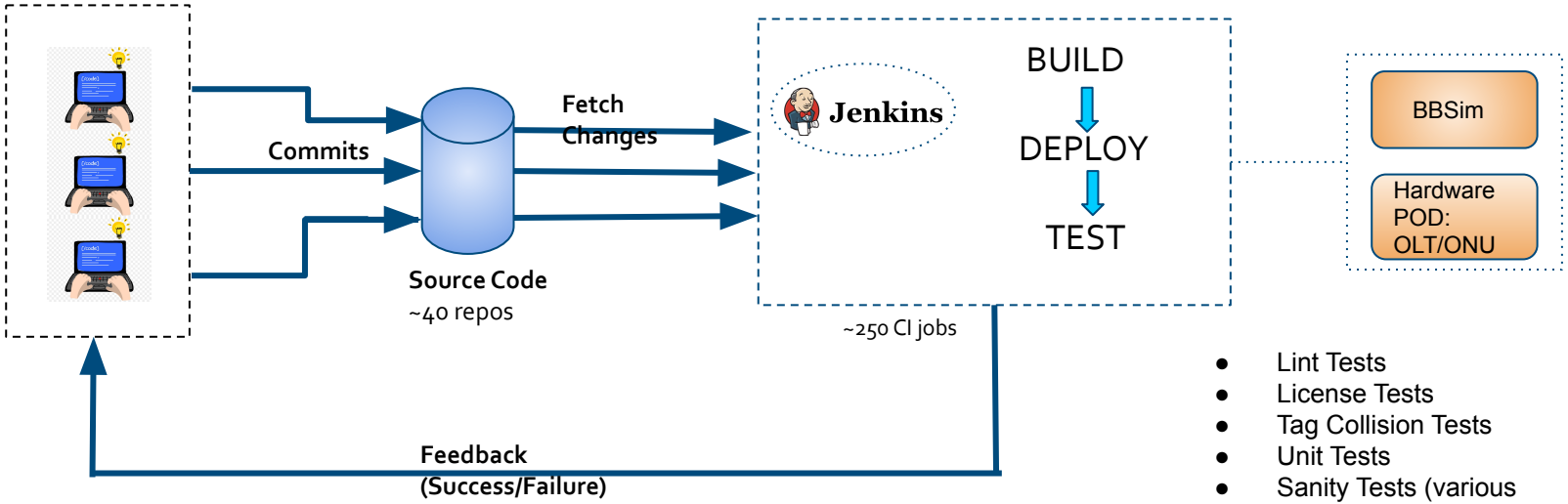
SEBA/Voltha Common Test Automation Framework



- Written in Robot Framework/python
- Supports testing on hardware (OLT/ONU) and BBSim (simulated OLT/ONU)
- Supports testing of various workflows (ATT, DT , TT)
- Provides debug/logging capabilities on tests
- Supports component level and system level tests

Voltha-system-tests repo

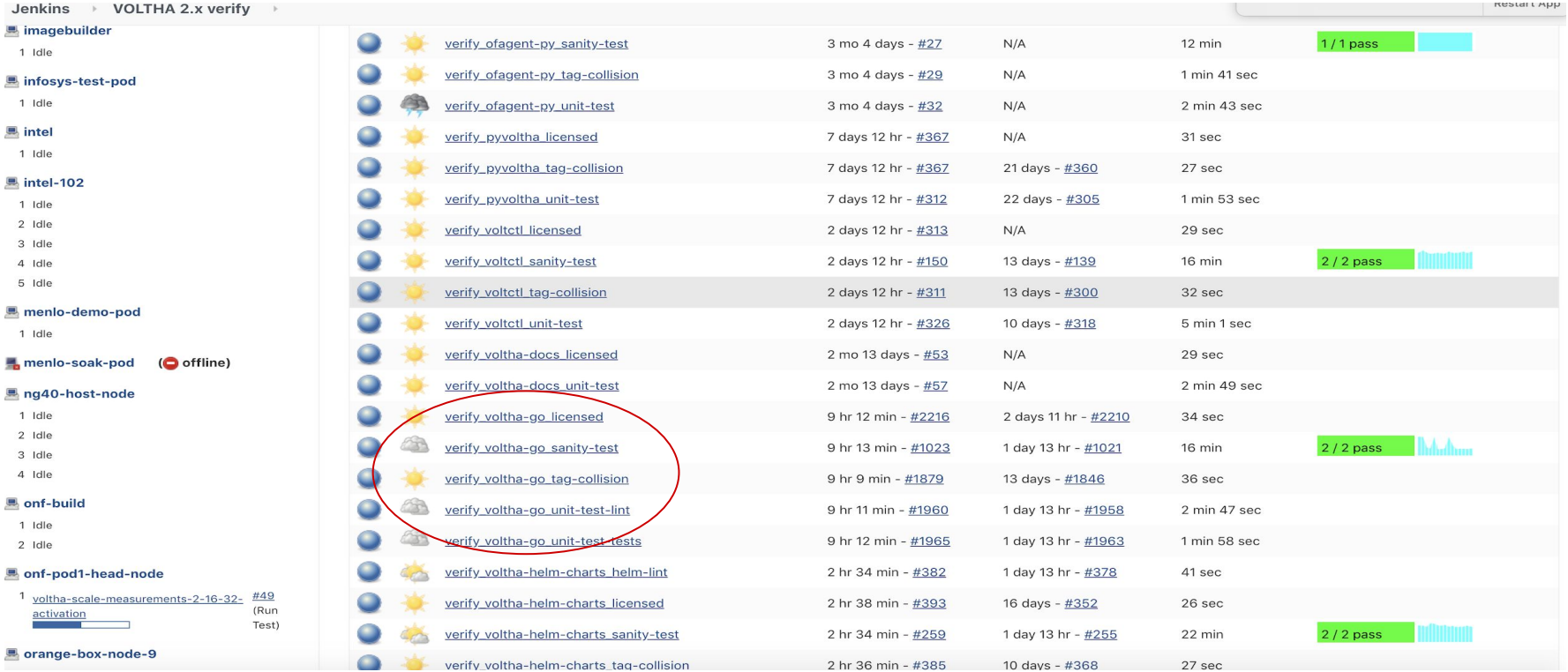
Patch Set Verifications - CI



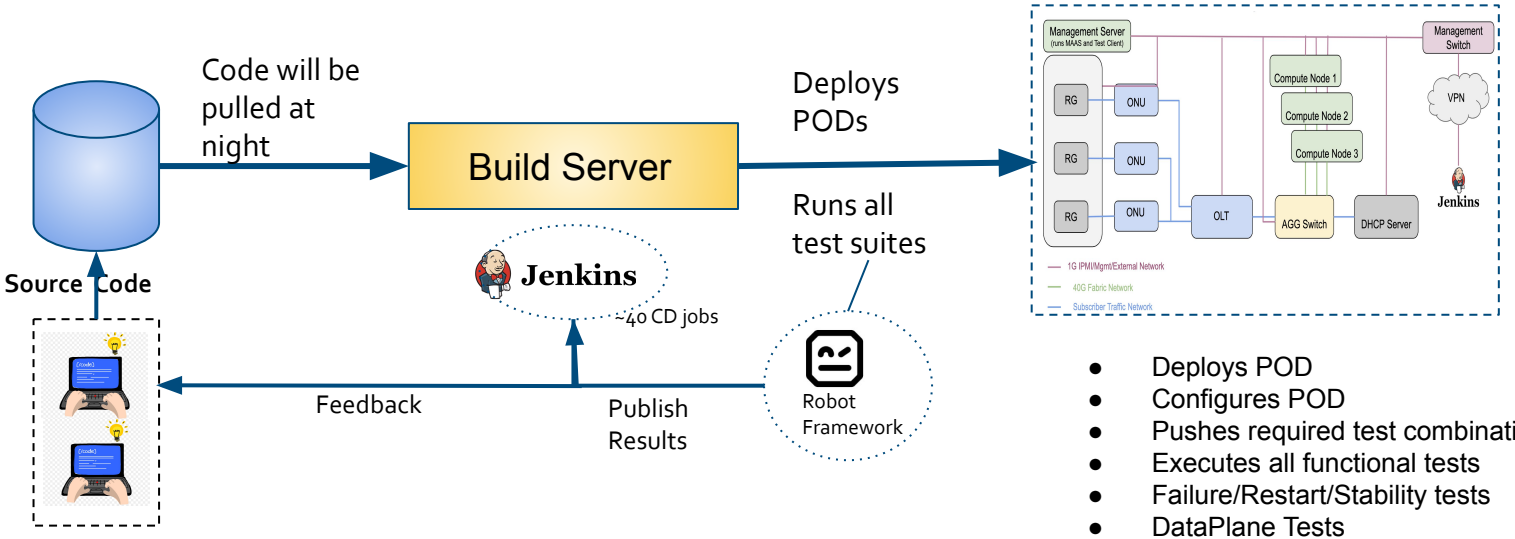
- Lint Tests
- License Tests
- Tag Collision Tests
- Unit Tests
- Sanity Tests (various workflows)
- Functional/Integration Tests

- Maintains developer velocity
- Faster feedback loop

Example - CI jobs from SEBA/Voltha

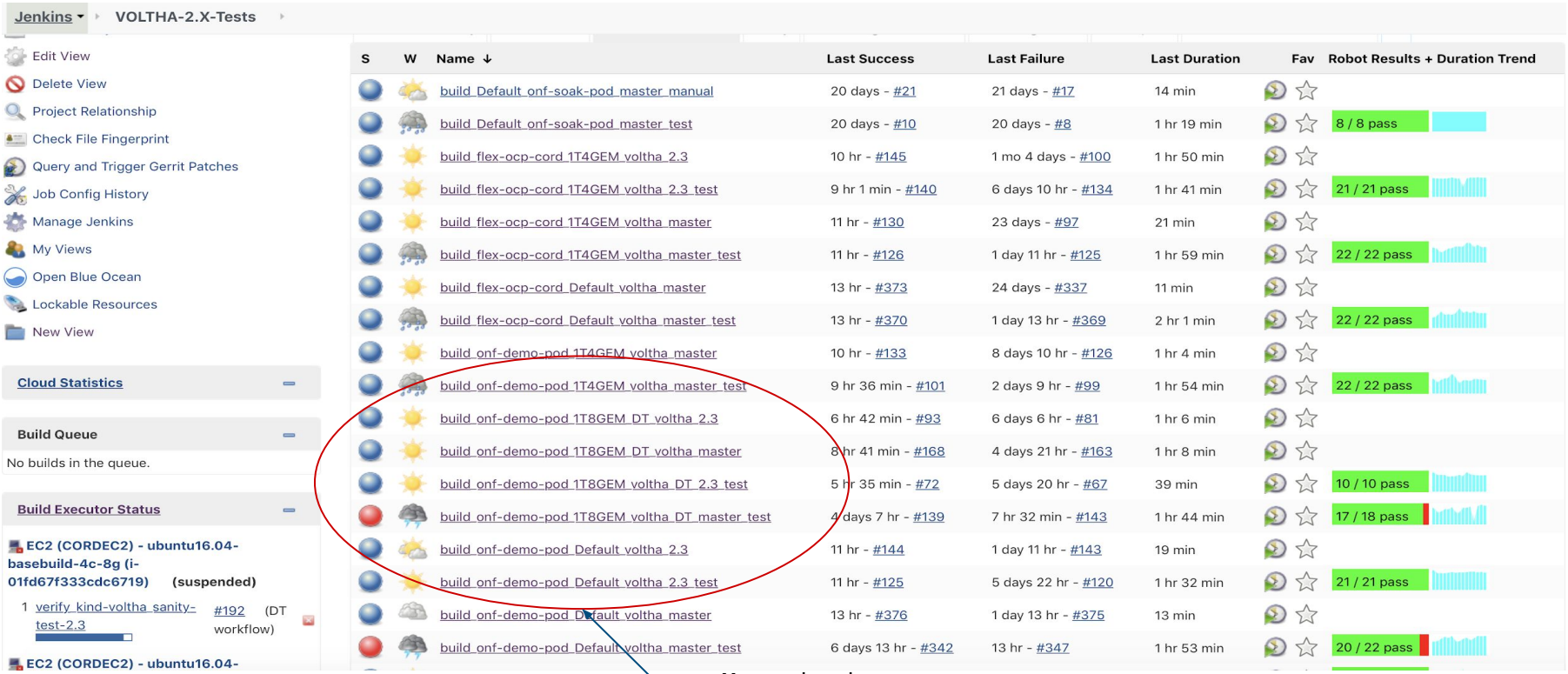


Nightly Jobs - CD



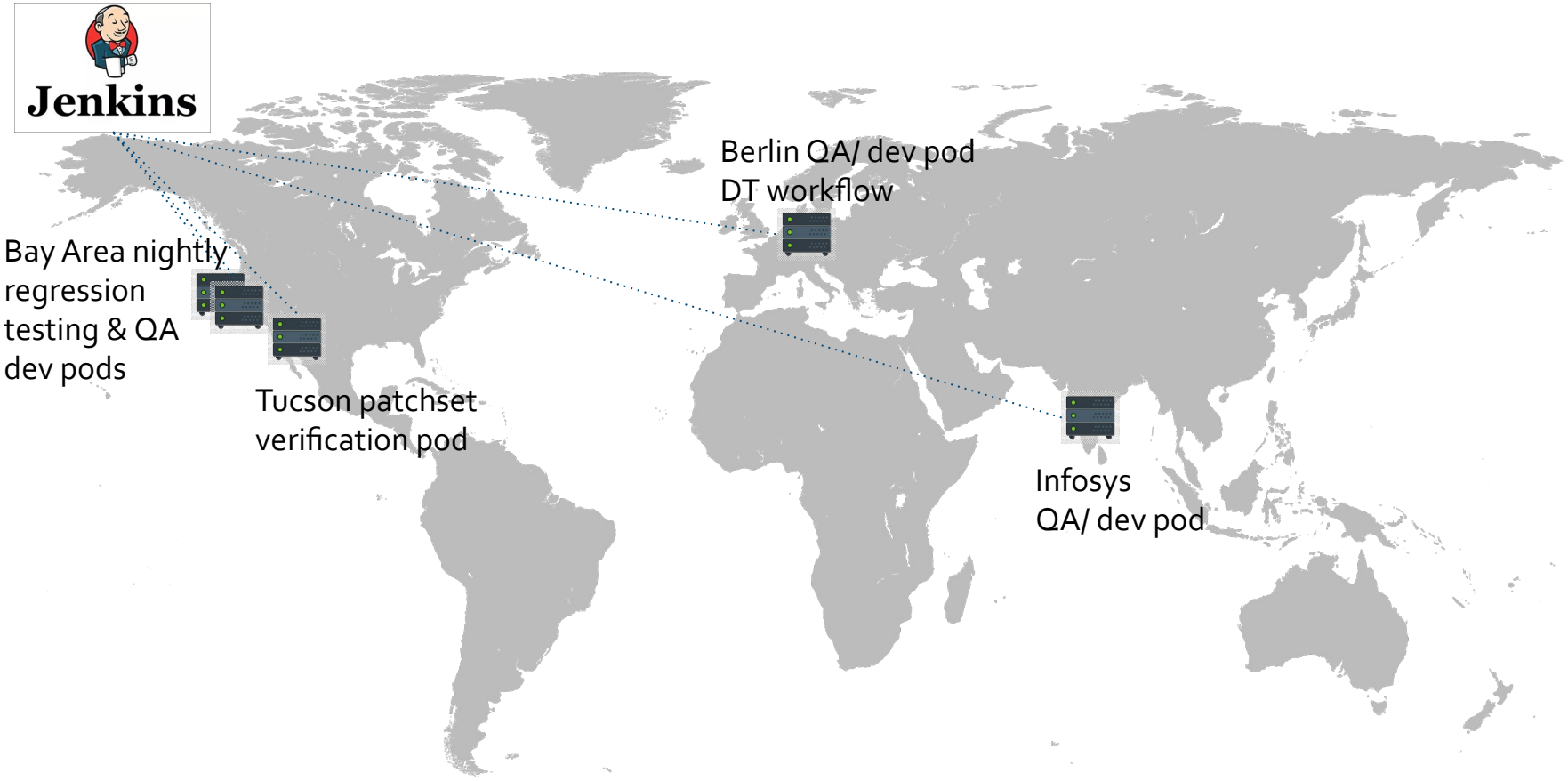
- Deploys POD
- Configures POD
- Pushes required test combinations
- Executes all functional tests
- Failure/Restart/Stability tests
- DataPlane Tests

Example - CD Jobs in SEBA/Voltha



- Master, released
- QoS Profiles
- Operator workflows

Hardware Test Pods

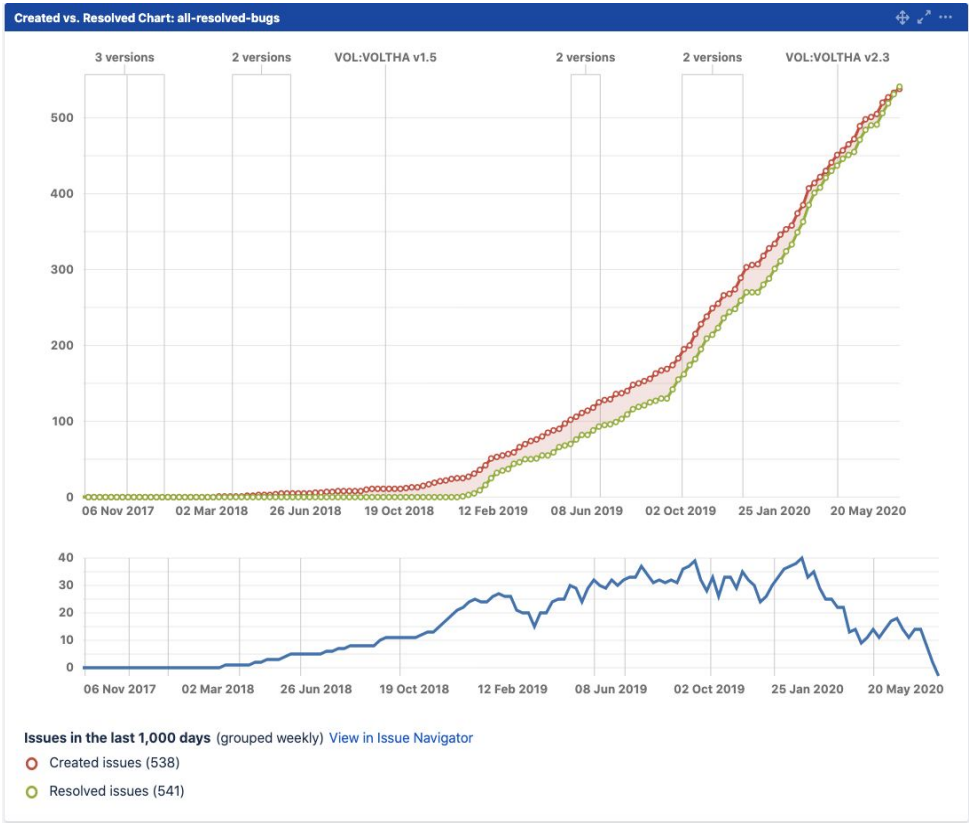


Various types of testing performed in SEBA/Voltha

- Unit testing
- Component/Functional verifications
  - Alarms, RedFish etc
- Use Case/End user validations
  - End-end integration, Stability, Failure/Recovery, DataPlane, Error checks
- Scale/Performance testing

Automation Status and Progress

Test type/Functional Area		% of Automated Tests		
		Voltha-2.2	Voltha-2.3	Voltha-2.4
System Level Functional Testing	AT&T workflow	15%	80%	90%
	DT workflow - FTTH	0%	35%	90%
	TT workflow	0%	0%	2%
Scalability		x	35%	60%
Stability		x	60%	90%
Alarms Functional		x	90%	90%
BBSim(Functional Coverage)		x	95%	95%
Device Management		x	x	80%
BAL upgrade 3.4		x	x	80%
Clustered ONOS		n/a	x	30%



What have we achieved so far with SEBA/Voltha?

- Aligned 11+ community organizations toward a set of goals
- Developed closed loop feedback between development and testing
- Reduced turn around time in finding/fixing bugs
- Common automation framework
  - Achieved multiple types of testing
- Created robust processes around CI/CD
  - Automated deployments on PODs around the community through ONF Jenkins
  - Keeps everyone up to date, increasing velocity

What's Missing and Next Steps?

- Continuous Deployment (reduce time to production)
  - Production style deployments (hardware)
  - Deploy staging pods continuously and run production payloads
  - Monitoring tools/systems for feedback/disaster recovery
- Other types of testing
  - Soak/Stress
  - More Automation
    - Multi-clustered ONOS, Multi-OLT, TT workflow
  - Chaos monkey

Operator challenge: bringing open source to production quality

How does one ensure quality software in an open source based solution?

- How to organize a community spanning multiple companies with different processes?
  - *Align around the distributed devops model*
- How to have ownership of issues? How to quickly resolve issues?
  - *ONF drives teams called brigades*
- How to establish common code/testing principles/guidelines?
  - *With distributed devops approach, defined workflows/guidelines for coding/testing*
- How to create common automation framework required to tame complexity?
  - *Use of powerful open source tools such as Robot Framework and Jenkins*
- How to include real world production traffic which is required to get to production level?
  - *Through use of common pods(share staging/field PODs) and CI/CD process*

Summary

- Community testing of open source solutions to production level is difficult, but can be achieved.
- ONF provides the focus needed to coordinate multiple organizations.
  - ONF community iterating on Distributed Devops practice
  - Common tooling and extensive automation is key to shortening test cycles and improving quality.
- Good progress, but more work needed to get to continual deployment models.

Join ONF Community, contribute and shape the project

Thank You

2020