ONAP

OPEN NETWORK AUTOMATION PLATFORM

ONAP Developer Typical Setup

2017 July 24-26 ONAP Virtual Developers Event

Gary Wu <gary.i.wu@huawei.com>

Daniel Rose <DR695H@att.com>

Victor Morales <victor.morales@intel.com>

Getting Started with ONAP Development

- How to get involved?
  - LF ID, mailing lists, Jira, wiki
- How to get and build the code?
  - git clone, building Java (maven, nexus, O-Parent), building Docker, bootstrap tools (vagrant-jenkins, vagrant-onap)
- How to run, test, create test cases?
  - UT, CSIT, E2E
- How to submit code changes?
  - Gerrit

Caveats

- Each ONAP project has different requirements and build procedures
- Tools and documentation are work in progress
  - Current state may differ from eventual goal
- We need your help!
  - Please contribute your expertise to help make ONAP easier for beginners

Goals of This Session

- For beginners
  - What a typical development flow looks like; what common steps are required
  - What tools and documentation are available to help
- For ONAP project teams
  - How your project differ from the "typical flow"
  - How you can help provide additional onboarding instructions specific to your project

Key ONAP Tools

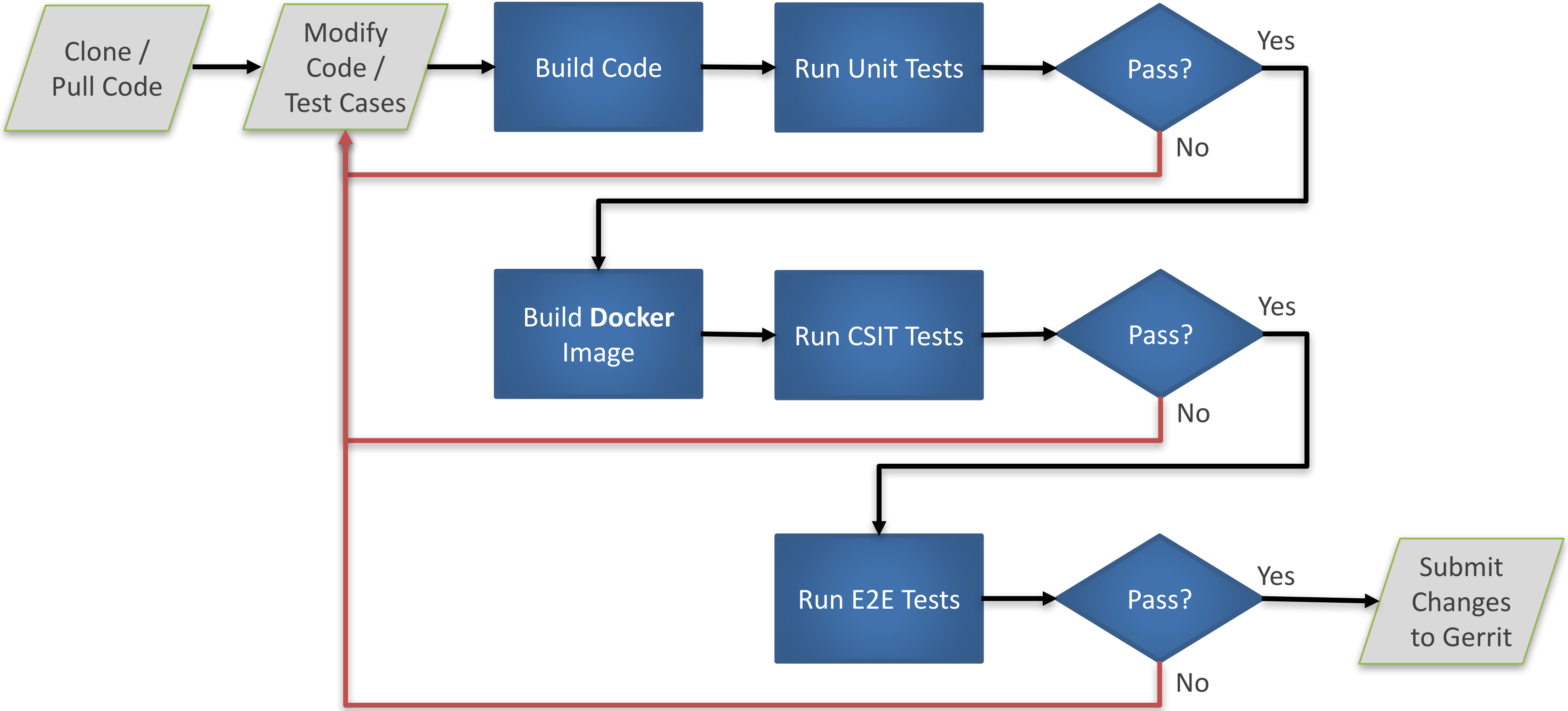
Getting Involved

- Start at the Wiki
  - https://wiki.onap.org/display/DW/Getting+Involved
- Get LF ID
  - https://identity.linuxfoundation.org/
- Subscribe to mailing lists
  - https://wiki.onap.org/display/DW/Mailing+Lists
- Track Issues with Jira
  - https://wiki.onap.org/display/DW/Tracking+Issues+with+JIRA

Environment Setup

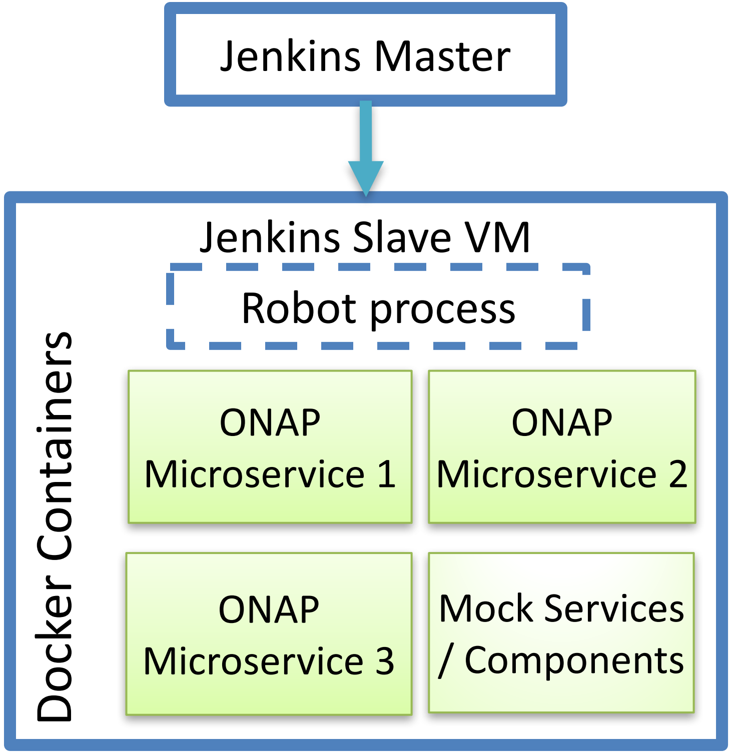
- Install git, java, maven, python, git-review
- Setup Maven settings.xml for Nexus location
- https://wiki.onap.org/display/DW/Setting+Up+Your+Development+Environment
- Clone the code
  - git clone ssh://USERNAME@gerrit.openecomp.org:29418/<REPONAME>
- Run Maven build
  - mvn clean install

Typical Development Flow (Ideal)



ONAP CSIT Infrastructure

- Test Suites written using Robot Framework
- Microservices run via docker
- Mock southbound services
  - e.g. mock VIMs or SDN controllers
- Robot test suites executed via Jenkins
  - Can also be run manually in developer’s environment
- https://wiki.onap.org/display/DW/Creating+a+CSIT+Test



Building Code / Running UT

- Run Maven build (Java)
  - mvn clean install
  - Projects should be structured so that the above also runs the UT test cases
- For non-Java projects, see individual project documentation
  - Compilation and UT may require separate commands
  - Recommend setting up a POM file so artifacts can be versioned and deployed in Nexus

Building Docker Images

- Different ONAP projects have different ways to build their docker images
  - Some use Maven (POM) plugins, some use shell scripts, some both
- See Bootstrap Tools (later slides) for information on building specific docker images
  - Can also see JJB definitions in ci-management repo
- Future goal: standardize the docker image build process

Running CSIT Test Plans

- CSIT test plans located in integration repo
  - [integration.git]/test/csit/
  - Use the test/csit/run-csit.sh script
    - Can be run in the developer’s local environment
- Currently there are no CSIT tests implemented
  - Projects are required/encouraged to develop CSIT test cases
- https://wiki.onap.org/display/DW/Creating+a+CSIT+Test

Running E2E Test Cases

- E2E tests located in testsuite repo
- Performs a full install of ONAP using heat templates
  - High / specific OpenStack system requirements
- Future goals:
  - Make it easier for individual developers to run E2E test cases
    - In developer virtual labs or on local environments

Bootstrap Tools – vagrant-onap

- Deploy and build ONAP project modules into a development environment
  - Reduce the barrier of entry to allow new ONAP developers to ramp up on to active development quickly
  - Reduce the cost to the community in responding to simple environment setup questions faced by new developers
- https://wiki.onap.org/display/DW/ONAP+on+Vagrant

Bootstrap Tools – vagrant-jenkins

- Automatically set up a Jenkins instance with predefined jobs to build all ONAP java code and docker images
- Shows how to set up an environment that can successfully build ONAP code from scratch
- https://wiki.onap.org/display/DW/Vagrant+Jenkins

O-Parent

- Common default settings as a parent POM
  - Each project sets its POM parent to inherit the defaults from O-Parent
- Isolate common configurations across ONAP projects
  - Avoid duplicate/conflicting settings for each project
  - E.g. Nexus location, checkstyle settings, Sonar configuration

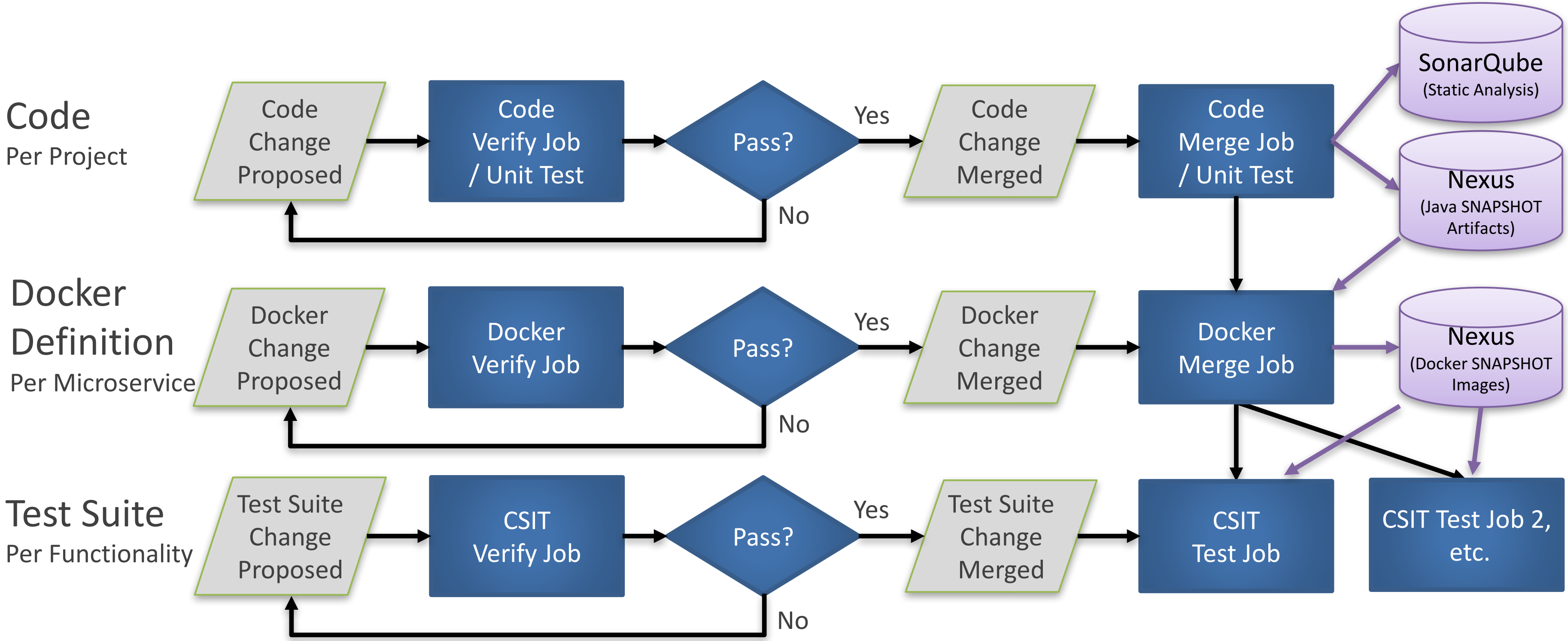
Project-Specific Development Guides

- https://wiki.onap.org/display/DW/Development+Guides
- Please review and add for your respective projects

Submitting Code Changes

- Add/modify test cases accordingly
- Make sure that all test cases pass
- Submit changes to gerrit
  - Use git-review tool
- Submitted changes will trigger Jenkins verification
- Once verified, project committer can review/merge

ONAP Gerrit / Jenkins Job Flow



Discussion / Q&A

- Common issues or pitfalls
- Where can we improve?