Radio Edge Cloud and Telco Appliance

Akraino Mini-Summit 2019.08.19-20

Paul Carver – paul.carver@att.com

AKRAINO EDGE STACK

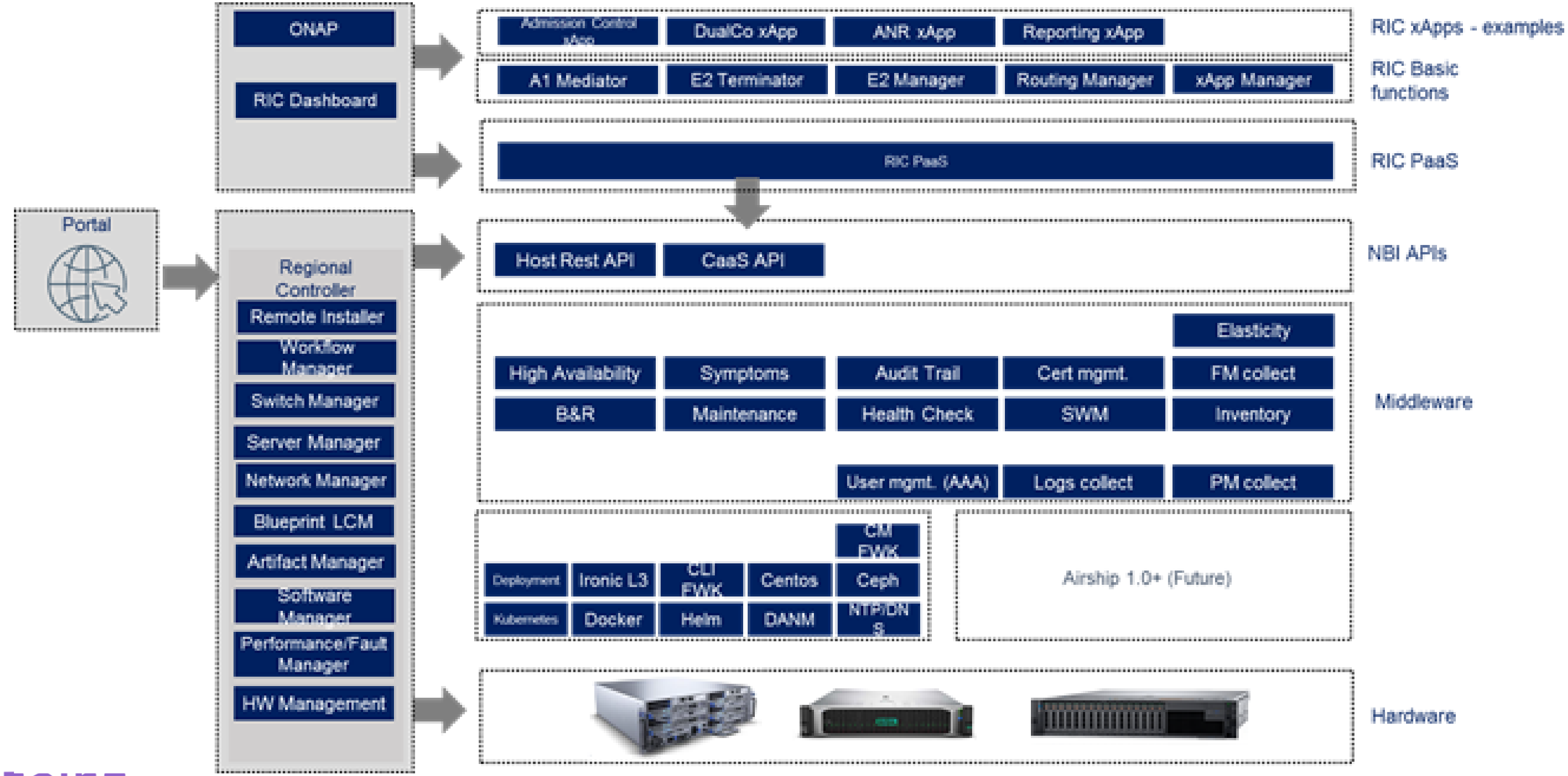
When in Doubt, Check the Wiki

- Most, if not all, of the information in this presentation is taken from pages that are already published to https://wiki.akraino.org
- Some helpful URLs
  - https://wiki.akraino.org/display/AK/Telco+Appliance+Blueprint+Family
  - https://wiki.akraino.org/display/AK/Radio+Edge+Cloud+Project+Meetings
  - https://wiki.akraino.org/display/AK/REC+Architecture+Document
  - https://wiki.akraino.org/display/AK/REC+Installation+Guide
  - https://wiki.akraino.org/display/AK/Gerrit+Code+Repository+Overview
  - https://wiki.akraino.org/display/AK/Radio+Edge+Cloud+Validation+Lab

Objectives of the Blueprint

- Fully automated simultaneous deployment and testing on multiple hardware platforms
  - Blueprint defines exact hardware configurations
  - Each hardware variant is deployed into a Continuous Deployment system that runs the full test suite
- Appliance model automates the installation, configuration and testing of:
  - Firmware and/or BIOS/UEFI
  - Base Operating System
  - Components for management of containers, performance, fault, logging, networking, CPU
- Application:
  - RIC is the application running on the REC appliance
  - Other appliances will be created by combining other applications with the same underlying components to create additional blueprints
  - Fully automated testing includes running full application test suite

Architecture



Portal

ONAP

RIC Dashboard

Regional Controller

Remote Installer

Workflow Manager

Switch Manager

Server Manager

Network Manager

Blueprint LCM

Artifact Manager

Software Manager

Performance/Fault Manager

HW Management

Admission Control xApp

DualCo xApp

ANR xApp

Reporting xApp

RIC xApps - examples

A1 Mediator

E2 Terminator

E2 Manager

Routing Manager

xApp Manager

RIC Basic functions

RIC PaaS

RIC PaaS

Host Rest API

CaaS API

NBI APIs

Elasticity

High Availability

Symptoms

Audit Trail

Cert mgmt.

FM collect

B&R

Maintenance

Health Check

SWM

Inventory

User mgmt. (AAA)

Logs collect

PM collect

Middleware

CM FWK

Deployment

Ironic L3

CLI FWK

Centos

Ceph

Kubernetes

Docker

Helm

DANM

NTP/DNS

Airship 1.0+ (Future)

Hardware

AKRAINO EDGE STACK

Components of the Blueprint

- **Components Used in Creation of the ISO Image**
- Build-tools: Based on OpenStack Disk Image Builder
- Dracut: Tool for building ISO images for CentOS
- RPM Builder: Common code for creating RPM packages
- Specs: the build specification for each RPM package
- Dockerfiles: the build specifications for each Docker container
- Unit files: the systemd configuration for starting/stopping services
- Ansible playbooks: Configuration of all the various components
- Test automation framework

Components of the Blueprint

- **Components Which Provide Additional REC Functionality**
- L3 Deployer: an OpenStack Ironic-based hardware manager framework
- Hardware Detector: Used to adapt L3 deployer to specific hardware
- Virtual installer: tooling to deploy REC on a VM (for testing only)
- North-bound REST API framework: For creating/extending REC APIs
- CLI interface
- AAA server to manage cloud infrastructure users and their roles
- Configuration management
- Container image registry
- Security hardening configuration
- A distributed lock management framework
- Remote Installer: Docker image used by Regional Controller to launch deployer

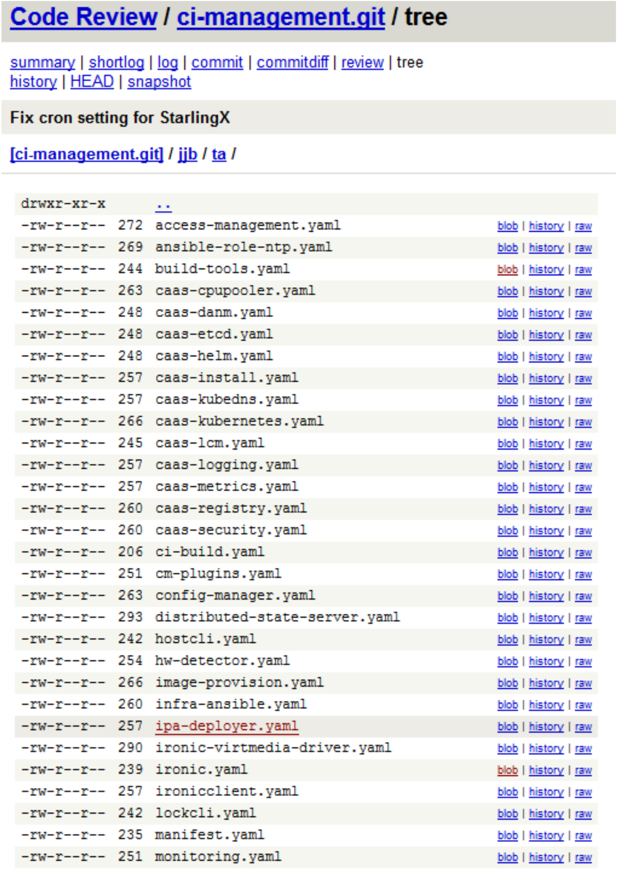
Components of the Blueprint

- **Upstream Components That Are Packaged Into REC with Configuration and Tuning:**
- CPU Pooler: Open Source Nokia project for K8s CPU management
- DANM: Open Source Nokia project for K8s network management
- Flannel: K8s networking component
- Helm: K8s package manager
- etcd: K8s distributed key-value store
- kubedns: K8s DNS
- Kubernetes
- Fluentd: Logging service
- Elasticsearch: Logging service
- Prometheus: Performance measurement service
- OpenStack Swift: Used for container image storage
- Ceph: Distributed block storage
- NTP: Network Time Protocol
- MariaDB, Galera: Database for OpenStack components
- RabbitMQ: Message Queue for Openstack components
- Python Peewee: A Python ORM
- Redis

AKRAINO
EDGE STACK

Continuous Integration

- https://jenkins.akraino.org/view/ta/



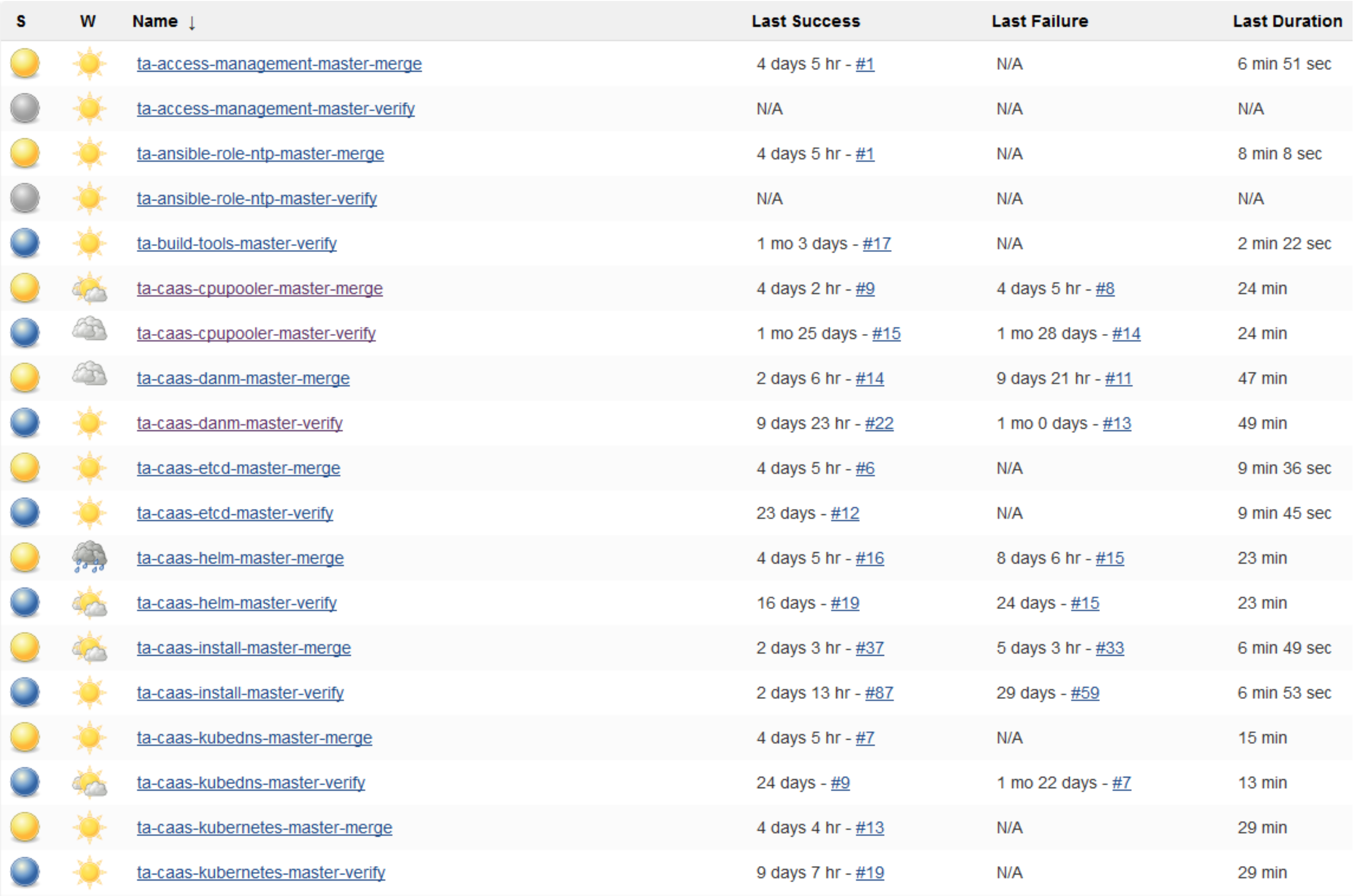
Code Review / ci-management.git / tree

summary | shortlog | log | commit | commitdiff | review | tree
history | HEAD | snapshot

Fix cron setting for StarlingX

[ci-management.git] / jjb / ta /

```
drwxr-xr-x       ..
-rw-r--r--  272  access-management.yaml              blob | history | raw
-rw-r--r--  269  ansible-role-ntp.yaml               blob | history | raw
-rw-r--r--  244  build-tools.yaml                    blob | history | raw
-rw-r--r--  263  caas-cpupooler.yaml                 blob | history | raw
-rw-r--r--  248  caas-danm.yaml                      blob | history | raw
-rw-r--r--  248  caas-etcd.yaml                      blob | history | raw
-rw-r--r--  248  caas-helm.yaml                      blob | history | raw
-rw-r--r--  257  caas-install.yaml                   blob | history | raw
-rw-r--r--  257  caas-kubedns.yaml                   blob | history | raw
-rw-r--r--  266  caas-kubernetes.yaml                blob | history | raw
-rw-r--r--  245  caas-lcm.yaml                       blob | history | raw
-rw-r--r--  257  caas-logging.yaml                   blob | history | raw
-rw-r--r--  257  caas-metrics.yaml                   blob | history | raw
-rw-r--r--  260  caas-registry.yaml                  blob | history | raw
-rw-r--r--  260  caas-security.yaml                  blob | history | raw
-rw-r--r--  206  ci-build.yaml                       blob | history | raw
-rw-r--r--  251  cm-plugins.yaml                     blob | history | raw
-rw-r--r--  263  config-manager.yaml                 blob | history | raw
-rw-r--r--  293  distributed-state-server.yaml       blob | history | raw
-rw-r--r--  242  hostcli.yaml                        blob | history | raw
-rw-r--r--  254  hw-detector.yaml                    blob | history | raw
-rw-r--r--  266  image-provision.yaml                blob | history | raw
-rw-r--r--  260  infra-ansible.yaml                  blob | history | raw
-rw-r--r--  257  ipa-deployer.yaml                   blob | history | raw
-rw-r--r--  290  ironic-virtmedia-driver.yaml        blob | history | raw
-rw-r--r--  239  ironic.yaml                         blob | history | raw
-rw-r--r--  257  ironicclient.yaml                   blob | history | raw
-rw-r--r--  242  lockcli.yaml                        blob | history | raw
-rw-r--r--  235  manifest.yaml                       blob | history | raw
-rw-r--r--  251  monitoring.yaml                     blob | history | raw
```

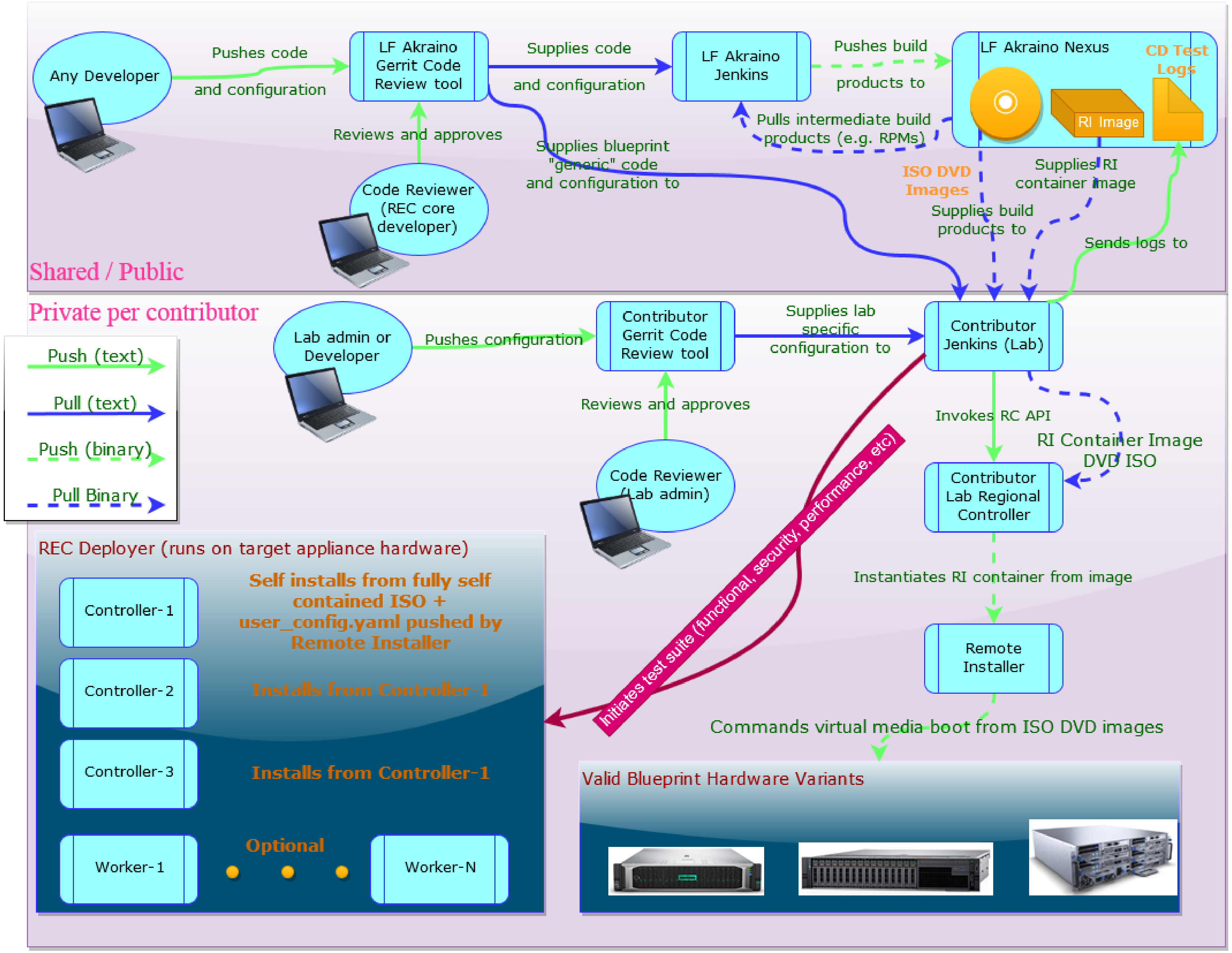


S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	ta-access-management-master-merge	4 days 5 hr - #1	N/A	6 min 51 sec
●	☀	ta-access-management-master-verify	N/A	N/A	N/A
●	☀	ta-ansible-role-ntp-master-merge	4 days 5 hr - #1	N/A	8 min 8 sec
●	☀	ta-ansible-role-ntp-master-verify	N/A	N/A	N/A
●	☀	ta-build-tools-master-verify	1 mo 3 days - #17	N/A	2 min 22 sec
●	⛅	ta-caas-cpupooler-master-merge	4 days 2 hr - #9	4 days 5 hr - #8	24 min
●	☁	ta-caas-cpupooler-master-verify	1 mo 25 days - #15	1 mo 28 days - #14	24 min
●	☁	ta-caas-danm-master-merge	2 days 6 hr - #14	9 days 21 hr - #11	47 min
●	☀	ta-caas-danm-master-verify	9 days 23 hr - #22	1 mo 0 days - #13	49 min
●	☀	ta-caas-etcd-master-merge	4 days 5 hr - #6	N/A	9 min 36 sec
●	☀	ta-caas-etcd-master-verify	23 days - #12	N/A	9 min 45 sec
●	🌧	ta-caas-helm-master-merge	4 days 5 hr - #16	8 days 6 hr - #15	23 min
●	⛅	ta-caas-helm-master-verify	16 days - #19	24 days - #15	23 min
●	⛅	ta-caas-install-master-merge	2 days 3 hr - #37	5 days 3 hr - #33	6 min 49 sec
●	☀	ta-caas-install-master-verify	2 days 13 hr - #87	29 days - #59	6 min 53 sec
●	☀	ta-caas-kubedns-master-merge	4 days 5 hr - #7	N/A	15 min
●	⛅	ta-caas-kubedns-master-verify	24 days - #9	1 mo 22 days - #7	13 min
●	☀	ta-caas-kubernetes-master-merge	4 days 4 hr - #13	N/A	29 min
●	☀	ta-caas-kubernetes-master-verify	9 days 7 hr - #19	N/A	29 min

Continuous Deployment Testing

- The basic structure of the CD part of the flow is an hourly check for new ISO images. If a new ISO build is found at https://nexus.akraino.org/content/repositories/images-snapshots/TA/release-1/images/ then it is downloaded and a deployment to bare metal OpenEdge19 cluster is attempted. If the ISO installs successfully, then the RIC install is attempted as a post-install job and some tests are automatically run.
- The future plan is to incorporate the RIC into a REC ISO, at which point the TA ISO and the REC ISO will be different. Right now there is only the TA ISO because we haven't built the CI jobs to integrate RIC and TA into a combined REC ISO.

CI/CD Pipeline


Any Developer
Pushes code and configuration
LF Akraino Gerrit Code Review tool
Supplies code and configuration
LF Akraino Jenkins
Pushes build products to
LF Akraino Nexus
CD Test Logs
RI Image
Reviews and approves
Code Reviewer (REC core developer)
Supplies blueprint "generic" code and configuration to
Pulls intermediate build products (e.g. RPMs)
ISO DVD Images
Supplies RI container image
Supplies build products to
Sends logs to
Shared / Public
Private per contributor
Push (text)
Pull (text)
Push (binary)
Pull Binary
Lab admin or Developer
Pushes configuration
Contributor Gerrit Code Review tool
Supplies lab specific configuration to
Contributor Jenkins (Lab)
Reviews and approves
Code Reviewer (Lab admin)
Invokes RC API
RI Container Image DVD ISO
Contributor Lab Regional Controller
Initiates test suite (functional, security, performance, etc)
Instantiates RI container from image
Remote Installer
Commands virtual media boot from ISO DVD images
REC Deployer (runs on target appliance hardware)
Controller-1
Self installs from fully self contained ISO + user_config.yaml pushed by Remote Installer
Controller-2
Installs from Controller-1
Controller-3
Installs from Controller-1
Optional
Worker-1
Worker-N
Valid Blueprint Hardware Variants

Test Suite Development

- https://jira.akraino.org/browse/REC
- The CD test suite is small but growing
- Building a comprehensive set of automated tests is currently the primary development focus of Radio Edge Cloud
- The full test suite will include upstream tests from the RAN Intelligent Controller project as well as tests from the Akraino Validation project and REC specific tests developed directly within the Akraino TA blueprint family and REC blueprint

AKRAINO EDGE STACK

AKRAINO
EDGE STACK

AKRAINO
EDGE STACK