Radio Edge Cloud and Telco Appliance

Akraino Mini-Summit 2019.08.19-20
Paul Carver – paul.carver@att.com
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
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/stoping 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/extend 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
# Continuous Integration

[https://jenkins.akraino.org/view/ta/](https://jenkins.akraino.org/view/ta/)

<table>
<thead>
<tr>
<th>S</th>
<th>W</th>
<th>Name</th>
<th>Last Success</th>
<th>Last Failure</th>
<th>Last Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ta-access-management-master-merge</td>
<td>4 days 5 hr - #1</td>
<td>N/A</td>
<td>6 min 51 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-access-management-master-verify</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-ansible-role-nbp-master-merge</td>
<td>4 days 5 hr - #1</td>
<td>N/A</td>
<td>8 min 5 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-ansible-role-nbp-master-verify</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-build-tools-master-verify</td>
<td>1 mo 3 days - #17</td>
<td>N/A</td>
<td>2 min 22 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-cpupower-master-merge</td>
<td>4 days 2 hr - #9</td>
<td>4 days 5 hr - #8</td>
<td>24 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-cpupower-master-verify</td>
<td>1 mo 25 days - #15</td>
<td>1 mo 28 days - #14</td>
<td>24 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-dimm-master-merge</td>
<td>2 days 6 hr - #14</td>
<td>9 days 21 hr - #11</td>
<td>47 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-dimm-master-verify</td>
<td>9 days 23 hr - #52</td>
<td>1 mo 6 days - #13</td>
<td>49 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-etcd-master-merge</td>
<td>4 days 5 hr - #6</td>
<td>N/A</td>
<td>9 min 36 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-etcd-master-verify</td>
<td>23 days - #12</td>
<td>N/A</td>
<td>9 min 45 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-helm-master-merge</td>
<td>4 days 5 hr - #16</td>
<td>8 days 6 hr - #15</td>
<td>23 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-helm-master-verify</td>
<td>16 days - #19</td>
<td>24 days - #15</td>
<td>23 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-instancetype-master-merge</td>
<td>2 days 3 hr - #37</td>
<td>5 days 3 hr - #33</td>
<td>6 min 49 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-instancetype-master-verify</td>
<td>2 days 13 hr - #57</td>
<td>29 days - #59</td>
<td>6 min 53 sec</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-kubernetes-master-merge</td>
<td>4 days 5 hr - #7</td>
<td>N/A</td>
<td>15 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-kubernetes-master-verify</td>
<td>24 days - #5</td>
<td>1 mo 22 days - #7</td>
<td>13 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-subernetes-master-merge</td>
<td>4 days 4 hr - #13</td>
<td>N/A</td>
<td>29 min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ta-cas-subernetes-master-verify</td>
<td>9 days 7 hr - #19</td>
<td>N/A</td>
<td>29 min</td>
</tr>
</tbody>
</table>
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.
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