

# Porting REC on aarch64

| Checkpoints | Subjects                                       | Staff placeholders                                                                |
|-------------|------------------------------------------------|-----------------------------------------------------------------------------------|
| 30 Sep 2019 | Arm-enabled TA code merged into TA blueprint   | <a href="#">Alex Antone Alexandru Avadanii</a> <a href="#">Valentin Radulescu</a> |
| 31 Oct 2019 | Arm-enabled SEBA code merged/upstreamed        | <a href="#">Cristina Pauna Ciprian Barbu</a>                                      |
| 31 Dec 2019 | Arm-enabled REC code merged into REC blueprint |                                                                                   |

## Overview of Porting Strategy

The build process is driven by the ci-management repo which controls the Jenkins jobs across all of the TA Git repositories (as well as the rest of the Akraino blueprints). The structure and type of TA repositories is described in detail at [Gerrit Code Repository Overview](#) but at a high level consists of repos that contain parts of the build system, repos that contain instructions for packaging specific components, repos that configure systems and a few other repos such as the Remote Installer and the Test Automation Framework. The main packaging strategy is to create Docker images which are then wrapped into RPM packages which are then combined into a QCOW2 image (for use by OpenStack Ironic to handle bare metal deployment) and finally combined into a DVD ISO image for virtual media booting by a management controller such as a BMC, iDRAC or iLO.

Important repos for the porting effort are:

- The hardware detector - <https://gerrit.akraino.org/r/gitweb?p=ta/hw-detector.git;a=tree>
- The build tools - <https://gerrit.akraino.org/r/gitweb?p=ta/build-tools.git;a=tree>
- The RPM builder - <https://gerrit.akraino.org/r/gitweb?p=ta/rpmbuilder.git;a=tree>
- The manifest - <https://gerrit.akraino.org/r/gitweb?p=ta/manifest.git;a=tree>

In order to port to AARCH64 it will be necessary to build AARCH64 versions of all the x86\_64 RPMs in [https://nexus3.akraino.org/service/rest/repository/browse/rpm.snapshots/TA/release-1/rpms/x86\\_64/](https://nexus3.akraino.org/service/rest/repository/browse/rpm.snapshots/TA/release-1/rpms/x86_64/) and this may require building different Docker images as well or possibly even downloading different upstream software. For example in the [Dockerfile that handles Prometheus](#) there are instructions to download the source code from GitHub using curl and then "make build". This may or may not produce a Docker image that can run on AARCH64. This needs to be validated for each component.

After the Docker images are built and packaged into RPMs they are pushed into Nexus3 by the individual CI jobs for each repo. Then the ta-ci-build job <https://jenkins.akraino.org/view/ta/job/ta-ci-build/configure> uses those RPMs to construct the ISO DVD image.

When the ISO DVD is booted via virtual media mount there are six stages of installation, starting with the first node of the cluster which is the one which boots from the ISO and then uses OpenStack Ironic to image the rest of the nodes in the cluster based on the information in user\_config.yaml. An essential part of the process is the recognition of what the specific hardware is, driven by the definitions in [https://gerrit.akraino.org/r/gitweb?p=ta/hw-detector.git;a=tree;f=src/hw\\_detector/hw\\_types;h=73a463fbd0d411f42f61caa8f7aea2e10b982b9;hb=HEAD](https://gerrit.akraino.org/r/gitweb?p=ta/hw-detector.git;a=tree;f=src/hw_detector/hw_types;h=73a463fbd0d411f42f61caa8f7aea2e10b982b9;hb=HEAD) so it will be necessary to add a type for each AARCH64 server that will be supported.

## RIC aarch64 porting

A JIRA ticket describing RIC aarch64 porting specifics is available at  [REC-80 - RIC port to AArch64](#)  . Please check this ticket for up to date information on aarch64 RIC porting.

## Child Pages

- [REC images](#)