

SEBA For ARM Documentation

- [SEBA for ARM Architecture Documentation](#)
 - [SEBA ARM Hardware](#)
- [SEBA for ARM Porting Iteration 1](#)
- [SEBA for ARM Porting Iteration 2](#)
 - [Software Platform](#)
 - [Base components](#)
 - [Software base](#)
 - [Main helm-charts git repository](#)
 - [Additional charts git repository](#)
 - [SEBA-in-a-Box and automation-tools](#)
 - [Test framework - cord-tester](#)
 - [Deliverables and installation notes](#)
 - [Helm repositories](#)
 - [The git repositories](#)
 - [Installing SEBA-in-a-Box](#)
 - [Continuous Integration work for SEBA on ARM](#)
 - [Hardware setup](#)
 - [Jenkins pipeline](#)
 - [Jenkins jobs](#)

There has been extensive work to port and validate SEBA running on ARM hardware, as described here.

The work for porting SEBA on ARM has started even before the SEBA Blueprint was made part of the Telco Appliance Blueprint Family and it did not target the TA Platform at that time. For this reason, the information captured in this page is not dependent or heavily tailored to using the TA Platform. However, since this is the only effort for porting SEBA on ARM architecture, and the work happens almost entirely in the context of Akraino, it made a lot of sense to have the information under the SEBA Blueprint wiki page.

SEBA for ARM Architecture Documentation

Please refer to the [SEBA Blueprint Architecture Document](#) for a high level architectural view of the SEBA Blueprint. The same principles and considerations apply here as well.

For a broader, more general view of the SEBA architecture, HW/SW requirements and usecases, please visit the upstream Opencord resources:

- [ONF SEBA](#)
- [Opencord SEBA Wiki](#)
- [CORD Guide - Prereqs](#)

The same principles and guidelines apply for running SEBA on ARM as well, with the mention that running on ARM equipped HW generally requires a more in-depth knowledge of the HW and SW platform.

SEBA ARM Hardware

Servers built with ARM Processors have been around for quite some time and there are a few different offerings on the market. However, for the effort of porting SEBA on ARM, the work has focused primarily on using just a few of them, based on commercial availability, costs and performance:

- [Gigabyte™ R120-T32](#) with [Marvell® ThunderX® CN8890 Processors](#)
- [Gigabyte™ R281-T91](#) with [Marvell® ThunderX2® CN9975 Processors](#)
- [Lenovo™ ThinkSystem HR330A](#) with [Ampere eMAG™ HR330A Processors](#)

SEBA for ARM Porting Iteration 1

The initial effort for porting SEBA on ARM started officially in Akraino in early 2019. At that time, the upstream Opencord project had released CORD Platform 6.1 and SEBA 1.0 as described in the [CORD 6.1 Guide](#).

Find more information in the related sub-pages:

- [SEBA on ARM - porting and upstreaming work](#)
- [SEBA 1.0.0 for ARM Installation instructions](#)

SEBA for ARM Porting Iteration 2

The latest effort for porting SEBA on ARM has since moved away from the old CORD Platform 6.1 and SEBA 1.0 because there was no more real interest in the upstream Opencord project to maintain them, and at the same time the operators were pushing for adding more functionality and make SEBA better in quality and production ready.

Ever since late 2019, Opencord has started working on SEBA 2.0, having realeased a [SEBA 2.0-alpha](#) based on [CORD Platform 7.0](#). The major component of SEBA, VOLTHA, has also been bumped from v1.6 to v1.7.

The [SEBA 2.0-alpha release notes](#) lists the major changes from the previous version.

However, the SEBA 2.0-alpha release was not considered stable enough at the time of the release, so the development work has continued upstream until a more stable, development version could be used as a reference for porting.

Software Platform

During the porting effort, SEBA has been verified almost exclusively using IEC Type 2 Platform. You can read more about the architecture and installation notes from the wiki pages:

- [Integrated Edge Cloud \(IEC\) Blueprint Family](#)
- [IEC Type 2 for Integrated Edge Cloud \(IEC\) Blueprint Family](#)
- [IEC Type1&2 Installation Guide for R2](#)

More recently, the work has also continued for verifying SEBA on a TA cluster running on the 3-node [Ampere Pod 1](#) in the [Akraio Community Lab at UNH-IOL](#).

You can get more information about the TA Platform from the wiki pages below. Note that TA is most commonly associated with running [Radio Edge Cloud \(REC\)](#) Blueprint, but the TA Platform is common for both REC and SEBA.

- [Telco Appliance Blueprint Family](#)
- [Porting REC on aarch64](#)
- [REC Installation Guide](#)

Note that at the moment, the work for verifying SEBA on ARM on TA is still in progress.

Base components

For this second iteration, the main base SW components that have been verified are as follows:

Component	Version	Comments
Base OS	Ubuntu 18.04 for aarch64	CentOS should also work when deploying SEBA on TA for example
Linux Kernel	4.15.0-99-generic #100-Ubuntu	
Kubernetes	1.15.0-00	
Docker	18.06.1~ce~3~0~ubuntu	
helm	v2.12.3	Client: &version.Version{SemVer:"v2.12.3", GitCommit:"eecf22f77df5f65c823aacd2dbd30ae6c65f186e", GitTreeState:"clean"} Server: &version.Version{SemVer:"v2.12.3+unreleased", GitCommit:"cdc09e54bca50a53c1e54f0e67b15a0913f5af1d", GitTreeState:"dirty"}

Software base

The 2nd iteration of the SEBA for ARM port uses a snapshot of the master (development) branch of the various OpenCORD components, which was taken at a point in time where it was stable enough to be considered for porting.

The porting was done on several key components from OpenCORD.

Main helm-charts git repository

OpenCORD stores all the helm charts components in the [helm-charts](#) repository. This allows keeping the charts under version control and tracking changes by the developers.

The porting started out from this repository, which has been forked to the [iecedge](#) github account, at some point in the beginning of March 2020. In order to distinguish the [iecedge](#) work from the upstream branches, everything is separated on the [cord-7.0-arm64](#) branch, as indicated. The base reference is commit [6622ff3](#).

[iecedge/helm-charts - cord-7.0-arm64](#)

It corresponds roughly to [cord-platform-7.2.3](#) and [seba-2.0.0-alpha1](#) but for specific chart versions and application versions you can consult the following spreadsheets from the Akraio Jira:

[SEBA-2.0 for ARM64 - chart versions](#)

In summary, here are the most important components and their chart and application versions:

Component	Chart version	App Version	Comments
etcd-operator	0.8.3	0.9.3	
kafka	0.13.3	5.0.1	
zookeeper	1.2.0	v3	
ONOS	2.0.1-b2	2.2.1-b5	
VOLTHA	1.0.9	1.7.0	
xos-core	3.0.13	3.3.1	xos-tosca version 1.3.0
seba-services	1.4.4	N/A	see spreadsheet above for more details
att-workflow	1.3.0	N/A	att-workflow driver chart version 1.1.2 appVersion 1.2.3

Additional charts git repository

Besides the CORD related component charts, there are other charts that had to be ported to make SEBA work on ARM architecture.

These components have been divided into two categories and the chart code has been placed in the helm-k8s-charts git repository:

<https://github.com/iecedge/helm-k8s-charts>

There are two categories, stable and incubator.

The stable category refers to the Kubernetes specific applications and is generally added by default when a Kubernetes cluster is created.

The incubator category refers to the kafka and zookeeper components, where "incubator" is the designation for the helm repository in Opencord parlance.

The table below lists the components by categories:

Component	Chart version
Incubator	
Zookeeper	0.13.3
Kafka	1.2.0
Stable	
etcd-operator	0.8.3
elasticsearch	1.15.3
fluentd-elasticsearch	1.4.1
grafana	4.0.2
kibana	1.1.2
logstash	1.4.2
prometheus	9.3.1

SEBA-in-a-Box and automation-tools

During the porting phase, it was decided to use [SEBA-in-a-Box](#) in the daily work, because it offers better view of the components involved and things that don't work. This has greatly helped with the porting work.

At the same time, SiaB is the only way that Opencord tests SEBA as a whole in daily jobs (see [Opencord Jenkins CI - SiaB Jobs](#)), and so it made sense to adopt the same tools and port them to ARM.

The main repository used for SiaB is the Opencord [automation-tools](#). This has been forked to iecegedge github account, also under the cord-7.0-arm64 branch. The base reference is commit [5220aeb](#).

[iecedge/automation-tools - cord-7.0-arm64](#)

Test framework - cord-tester

You can read more about the Opencord test framework at these locations:

- [CORD Guide - Testing CORD](#)
- [System Test Guide](#)
- [Automation Framework \(CORD\)](#)

For the purpose of validating SEBA for ARM in Akraino, only a part of the testing framework has been re-utilized. Because testing a real usecase requires a complex setup and dedicated PON Hardware, for now only SEBA-in-a-Box has been ported and adapted to work in the Akraino CI.

This means adopting [cord-tester](#) and reproducing the [SIAB stable Jenkins Job](#).

The [cord-tester](#) was forked to iecedge github account, under cord-7.0-arm64 branch. The base reference commit is [5ce76d4](#).

[iecedge/cord-tester - cord-7.0-arm64](#)

Deliverables and installation notes

Helm repositories

SEBA is generally deployed using 3 set of charts, here referred to as "combined charts" or "release charts":

- cord-platform
- seba
- workflow charts (e.g. att-workflow)

The usual method is to add a helm repository and then install these one by one, as described in the [SEBA Blueprint Installation Guide](#).

For the 2nd iteration of the SEBA porting effort, a helm repository has been created using github.com pages. You can view the helm repo index.yaml here:

<https://iecedge.github.io/helm-charts/cord/index.yaml>

Additionally, another two helm repositories have been created for storing charts for components not associated with Opencord. This helps to keep the main CORD specific repository free from generic, Kubernetes and third party applications. You can view the indexes of these two helm repositories here:

<https://iecedge.github.io/helm-k8s-charts/incubator/index.yaml>

<https://iecedge.github.io/helm-k8s-charts/stable/index.yaml>

Step by step installation instructions as follows:

- Add the CORD repository and updated indexes

```
helm repo add cord https://iecedge.github.io/helm-charts/cord
helm repo update
```

- Install the CORD platform

```
helm install -n cord-platform --version 7.2.3-dev-arm64 cord/cord-platform
```

- Install SEBA

```
helm install -n seba --version 2.0.0-dev-arm64 cord/seba
```

- Install workflow

```
helm install -n att-workflow --version 1.3.0 cord/att-workflow
```

The git repositories

It is possible to install SEBA as individual components as well, using the forked [iecedge/helm-charts - cord-7.0-arm64](#).

Note that at this moment, this is the only possible method to install SEBA for ARM by individual components. In contrast, the Opencord also delivers the individual components as packaged helm charts in their <https://charts.opencord.org/> helm repo. However, using the git repository has the same effect, so functionality is not lost.

To begin, you need to clone the git repository:

```
git clone https://github.com/iecedge/helm-charts.git -b cord-7.0-arm64
git clone https://github.com/iecedge/helm-k8s-charts.git -b master
```

You can follow instructions on the [SEBA Installation Guide](#) for installing the individual components. Note that instead of installing the components from the helm repository, you need to install them from the git repository, by specifying the absolute or relative path to the respective directories.

Installing SEBA-in-a-Box

If you decide to install SEBA-in-a-Box for arm64, you need to clone the forked [iecedge/automation-tools](#) and run a few commands.

Note that during the porting work, the HW setup implied using ieceedge 3-node clusters, so the charts and commands have been tailored to work on 3 nodes, rather than a single node like in upstream Opencord. The same should work just as well on a single node, but it hasn't been verified recently.

If needed, check the information on [Integrated Edge Cloud \(IEC\) Blueprint Family](#) and the [IEC Type1&2 Installation Guide for R2](#).

Also note that automation-tools implies an Ubuntu environment, Ubuntu 18.04 is verified and recommended. Alternatively, you can try running on different distributions, as long as you modify the seba-in-a-box/Makefile accordingly and make sure the necessary dependencies are installed.

```
git clone https://github.com/iecedge/automation-tools -b cord-7.0-arm64
cd automation-tools/seba-in-a-box
# The env.sh prepares the environment in a way that seba-in-a-box/Makefile expects it to be
# and sets helm-charts to the ieceedge specific repo
source env.sh
make stable
```

Once the installation is finished, you can also try to run the SIAB tests:

```
make run-tests
```

Continuous Integration work for SEBA on ARM

As with the other Blueprints in Akraino, SEBA is also periodically verified for correct functionality in the [Akraino Jenkins](#) infrastructure.

For the moment, the CI jobs make use of the [IEC Type 2 Platform](#), deployed using the Fuel Installer. Alternatively, the IEC Type 2 Platform can also be deployed using the Compass Installer, which is not in the scope of this documentation.

Hardware setup

Please refer to the [Deployment Architecture](#) documentation for better understanding the HW setup and requirements.

At this moment, the Jenkins jobs for deploying and verifying SEBA on ARM, are running on a single POD, located in the ENEA Kista Lab, and connected to the Akraino Jenkins Master.

The jump host that connects to the Akraino Jenkins master is designated by the following Jenkins executor.

<https://jenkins.akraino.org/computer/prd-ubuntu1804-dev-96c-256g-1/>

There is also work in progress for adding another POD, the [ThunderX2 Pod 2](#) from the [Akraino Community Lab at UNH-IOL](#).

Jenkins pipeline

Currently, SEBA for ARM is verified as part of a bigger pipeline for IEC, which is located under the [IEC Jenkins project](#):

<https://jenkins.akraino.org/view/iec/job/iec-type2-fuel-baremetal-ubuntu1804-daily-master/>

Note that at this moment, only IEC Type 2 Fuel Baremetal Jenkins jobs support deploying and testing SEBA.

Deploying on the virtual PODs in the Akraino Jenkins brings little value, because of performance penalties.

Also, the Fuel Installer is the only that has been verified, but Compass should work as well. However, there are currently no Compass Baremetal jobs, so the CI needs to be restricted to Fuel baremetal installations for now.

Jenkins jobs

As you can see, the parent job runs a few different jobs:

- [iec-type2-deploy-fuel-baremetal-ubuntu1804-daily-master](#) - this deploys the IEC Type 2 Platform using the Fuel Installer and Ubuntu 18.04 as base OS
- [validation-enea-daily-master](#) - IEC Blueprint Validation as per Akraino requirements
- [iec-type2-install-seba_on_arm-fuel-baremetal-ubuntu1804-daily-master](#) - deploys SEBA-in-a-Box using automation-tools
- [iec-type2-test-seba_on_arm-fuel-baremetal-ubuntu1804-daily-master](#) - runs the SEBA-in-a-Box Robot tests

For the test-seba_on_arm job, the robot logs are collected and sent to the Jenkins master for aggregation and visualization.