EMBArk already has the build/installation well documented here: https://github.com/e-m-b-a/embark/wiki/Installation

Here is the new documentation for offline worker node deployment which is available in ./Documentation/worker.md:

Offline worker configuration

As the offline worker has no internet access, dependencies have to be fetched on the host first. This is done using `apt-get download`. One should setup a virtual machine using the OS of the future offline worker, and fetch all dependencies using the `examples/setup_worker/host.sh` script.

The dependencies are:

- EMBA source code
- Docker package incl. dependencies (e.g. `iptables`)
- Docker compose package
- `inotify-tools` package
- `libnotify-bin` package
- Exported EMBA docker image (`embeddedanalyzer/emba`)
- External data (NVD Json data feeds, EPSS data)

Note: While EMBA uses a virtual environment for execution, no additional python packages are needed. Thus, the virtual environment is faked to minimize dependencies.

Dependencies

All dependencies should be provided to the offline worker using a mountpoint. The structure is as follows:

```
+- installer.sh
+- uninstaller.sh
+- emba.tar.gz                    # EMBA source code
+- pkg/                           # apt packages (incl. dependencies)
+- emba-docker-image.tar
+- external/
|  +- nvd-json-data-feeds
|  +- EPSS-data
+- test/
|  +- firmware.zip                # Optional: firmware for EMBA test run
|  +- run_emba_test.sh            # Optional: script for EMBA test run
+- emba_venv/
   +- bin/
      +- activate                 # empty file, fakes unused virtual environment
```

Worker setup

Once all dependencies are fetched, the `installer.sh` should be executed on each offline worker. It installs all the provided packages, and creates files where needed.

The result is a working EMBA installation in `/root/emba`, ready for use.

If the installation should be removed, run `./uninstaller.sh`.

Testing EMBA

An exemplary invocation of EMBA by EMBArk can be found in `./test/run_emba_test.sh`.