

# Writing xApps



## Obsolete Note

We are in the process of moving this xApp writing guide to <https://docs.o-ran-sc.org>.

Please refer to [App Writing Guide](#) for latest guide.

Logically, an xApp is an entity that implements a well-defined function. Mechanically, an xApp is a K8s pod that consists of one or multiple containers. In order for an xApp to be deployable, it needs to have an xApp descriptor (JSON) that describes the xApp's configuration parameters and information the RIC platform needs to configure the RIC platform for the xApp. The xApp developer will also need to provide a JSON schema for the descriptor.

In addition to these basic requirements, an xApp may do any of the following:

- Read initial configuration parameters (passed in the xApp descriptor).
- Receive updated configuration parameters.
- Send and receive messages.
- Read and write into a persistent shared data storage (key-value store).
- Receive A1-P policy guidance messages - specifically operations to create or delete a policy instance (JSON payload on an RMR message) related to a given policy type.
- Define a new A1 policy type.
- Make subscriptions via E2 interface to the RAN, receive E2 INDICATION messages from the RAN, and issue E2 POLICY and CONTROL messages to the RAN.
- Report metrics related to its own execution or observed RAN events.

The lifecycle of xApp development and deployment consists of the following states:

- Development: Design, implementation, local testing.
- Released: The xApp code and xapp descriptor are committed to LF Gerrit repo and included in an O-RAN release. The xApp is packaged as a Docker container and its image released to LF Release registry.
- On-boarded/Distributed: The xApp descriptor (and potentially helm chart) is customized for a given RIC environment and the resulting customized helm chart is stored in a local helm chart repo used by the RIC environment's xApp Manager.
- Run-time Parameters Configuration: Before the xApp can be deployed, run-time helm chart parameters will be provided by the operator to customize the xApp Kubernetes deployment instance. This procedure is mainly used to configure run-time unique helm chart parameters such as instance UUID, liveness check, east-bound and north-bound service endpoints (e.g., DBAAS entry, VES collector endpoint) and so on.
- Deployed: The xApp has been deployed via the xApp Manager and the xApp pod is running on a RIC instance. For xApps where it makes sense, the deployed status may be further divided into additional states controlled via xApp configuration updates. For example,
  - Running
  - Stopped