

4.0 DIDO Data Lifecycle Language (DDLL)

[Return to Part I](#)

The DIDO Lifecycle Language (DDLL) is responsible for controlling the Lifecycle of the DIDO instances distributed across the [Nodes](#) in the [DIDO Network](#). The DidoLL does not cover the [System Lifecycle](#) which is responsible for the system conception, design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out and disposal. Although there is some overlap, the System Lifecycle covers also covers the development of the [DIDO Platform](#) while the DidoLL only covers the deployment and execution of the DIDO software on the individual nodes.

Although there is no [requirement](#) for a DIDO to be implemented as a [Virtual Node](#) (i.e., [Virtual Machine \(VM\)](#) or [Application Container](#)), the basic steps would be the similar. McGee¹ has proposed the following steps for application containers (i.e., [Docker](#)).

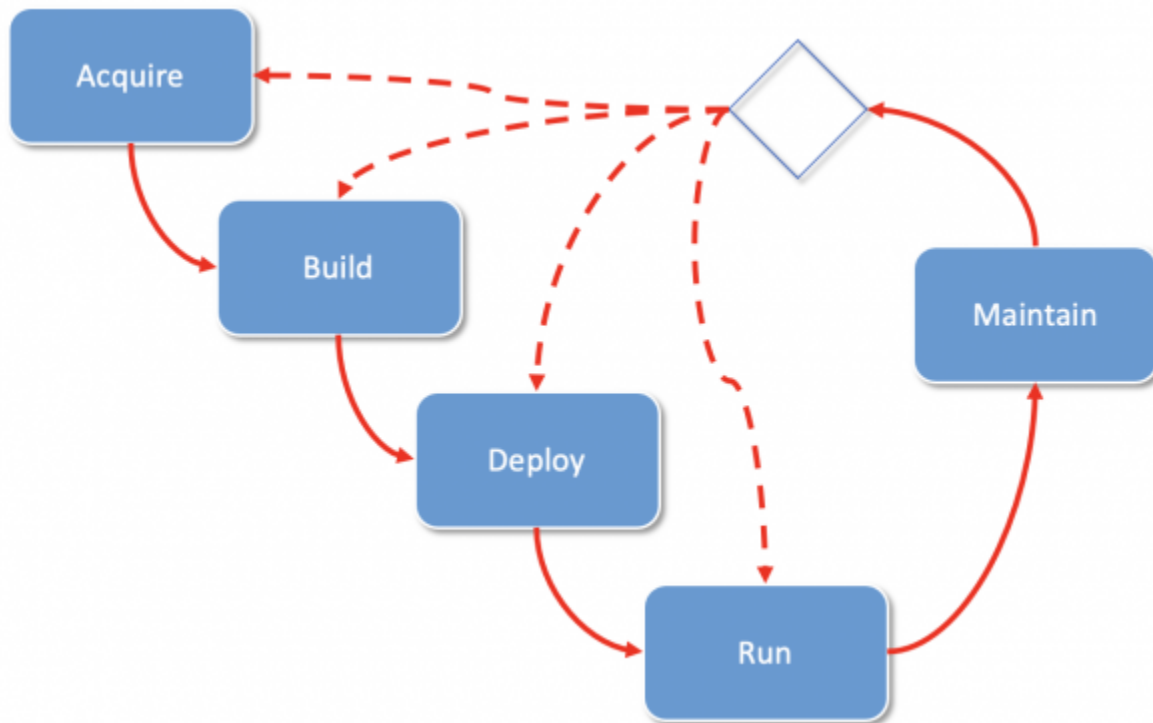


Figure 1: The 6 steps of a [container's](#) lifecycle.

1. **Acquire** - Gathering the products (i.e, requirements, laying out a system plan, hardware, software or content)
2. **Build** - Assemble the acquired products in to system that can be deployed.
3. **Deliver** - The assembled system is delivered for validation and verification
4. **Deploy** - is the distribution of the approved system to the initial set of nodes nd the verification and validation that the system id properly deployed.
5. **Run** - The deployed system is started and the nodes are working as a system. The operating states (starting, running, shutdown, etc) are reported.
6. **Maintain** - Anomalies and faults are detected, reported for further remediation.

- **Note:** Although the steps are presented as a linear progression from steps 1-6, the steps 3-6 (i.e., **Deploy**, **Run**, and **Maintain**) are repeated for every node within the DIDO network. If a flaw is repaired as a result of an anomaly or fault reported and a fix is identified, then the **Build** Step might need to be repeated. Sometimes, the flaw might require that the **Acquire** step be repeated.

The DidoLL needs to support the entire Lifecycle of the DIDO on each individual Node.

DDL Commands

- [4.1 NetworkProtocol](#)
- [4.1 PortNumberType](#)
- [4.2 Port](#)
- [4.3 Machine](#)
- [4.4 Resource](#)
- [4.5 Volume](#)
- [4.6 Virtual Machine](#)
- [4.7 Container](#)

1)
Jason McGee, IBM, [The 6 steps of the container lifecycle](https://www.ibm.com/blogs/cloud-computing/2016/02/08/the-6-steps-of-the-container-lifecycle/), 8 February 2016, Accessed: 29 March 2021, <https://www.ibm.com/blogs/cloud-computing/2016/02/08/the-6-steps-of-the-container-lifecycle/>

From:
<https://www.omgwiki.org/dido/> - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:04_dll:start&rev=1629217707

Last update: **2021/08/17 12:28**

