

Name	SO IDs	Involved Concept	Description
Executable Name (ID 1)	27767264, 31696439, 53691781, 44919972, 52120845, 47302658, 39240560, 64525899, 27612209, 31070671, 46715072, 43707770, 59076546, 54429751, 51768216, 41998862, 56938511, 41804297, 55562381, 37490324, 37301460, 33445718, 34534079, 56202090, 59330607	Maven, Docker	Maven and Docker depend on the executable name, which is specified in the pom.xml and in the Dockerfile. Specifically, Maven creates the executable name using the artifactID, version, and a packaging format, which defaults to JAR if not specified. Within the Dockerfile, the executable name is either used by the first argument of Dockers ADD or COPY command to add the executable to the file system of the container. Note that the executable name is usually part of a file path, since Maven stores the executable name in the target directory.
Internal Docker Dependency (ID 2)	59747862, 51679363, 40522967, 47302658, 39240560, 28907975, 64525899, 46057625, 27612209, 31070671, 46715072, 43707770, 34170216, 55752013, 59076546, 54429751, 51768216, 61019493, 41998862, 49941955, 56938511, 41804297, 48359972, 55562381, 37490324, 52069406, 37301460, 58853372, 39217402, 26937810, 51200107, 43235946, 53407287, 41181159, 33445718, 38222513, 34534079, 56202090, 59330607	Docker	Within Docker there is often an internal dependency between Docker's ADD/COPY and ENTRYPOINT/CMD commands, e.g., when both commands specify the executable name inside the container. Specifically, the internal Docker dependency exists between the second argument of Docker's ADD/COPY command and one part of the ENTRYPOINT/CMD command.
Port (ID 3)	46057625, 55752013, 56938511, 54145220, 44667063, 56441497, 53407287, 53089486, 44667063, 38222513, 56753810, 59330607, 51085037, 65183932, 43720339, 40905761, 42567475, 37100358, 51097652	Docker, Docker-Compose, Spring Boot	Dockerized Spring Boot applications usually exhibit multiple configuration options, which specify the applications reachable port. Due to the nature of Docker containers, all ports that are available from outside the container have to be explicitly exposed. In this case, Docker Compose bundles different containers together in a new one, which means the port has to be exposed again. That is, the port is defined at different places, such as in the Dockerfile, docker-compose.yml, and application.yml.
Database Credential (ID 4)	58880998, 54145220, 56441497, 56202090, 56753810	Docker-Compose, Spring Boot	Multi-container applications often incorporate databases, such as a MySQL or PostgreSQL. To access those databases, credentials are specified at different configuration artifacts, such as in Spring Boot's property file and in the docker-compose.yml.
Configuration Artifact (ID 5)	42208442, 27749193, 53691781, 34203179, 35132568, 42278216, 57581943, 33381235, 49395983, 47969389, 28078524, 65437831, 59747862, 38293073, 38692481, 40898608, 51679363, 40522967, 52892488, 60522767, 28907975, 56303459, 59076546, 52788472, 29480099, 30043872, 29905953, 42567475, 57048428, 31210973, 51097652	Docker, Maven, JSON	When Docker is used for deployment, often configuration files, such as the pom.xml and package.json, are copied from the local root directory to the host machine's file system to build and run the application in a Docker container. This means there is a configuration dependency between Docker's COPY/ADD command and the name of configuration artifact to be copied.