Features

- Configuration — A Configuration specifies how the container is configured (logging, security, data sources, location where to put deployables, etc).
  - Configuration properties — Properties to configure a container (request port, shutdown port, logging level, threads, etc)
  - Custom Configuration — A user-provided configuration to use instead of the default Cargo-provided ones
  - Local Configuration — A configuration for a container that is running on the local machine where Cargo is executing
    - Existing Local Configuration — Re-use an existing container installation
    - Standalone Local Configuration — Configures your container in a specific directory
      - Configuration files option — The configuration files are the files to add to your container's configuration. It is internally used by some containers and is accessible for you can to add extra files.
      - DataSource and Resource Support
        - Containers with DataSource and Resource support
  - Runtime Configuration — A configuration for a container that is already started
- Container — A top level interface wrapping a real physical container
  - Container Instantiation — Create a container instance
  - Local Container — A container executing on the machine where Cargo executes
    - Container Classpath — How to configure the executing container's classpath
      - Application Classpath — How to configure the classpath of the application without changing the container's own classpath
    - Container Start — Start a container that is not already running
    - Container Stop — Stop a running container
    - Container Timeout — Timeout after which the container start/stop is deemed failed
    - Embedded Container — A container executing in the same JVM where Cargo is running
    - Installed Container — A container installed on the machine where Cargo executes.
      - Installer — Installs a container
      - Passing system properties — How to pass system properties that will be available to the container while executing
  - Remote Container — A container that is already running on some machine (local or remote)
- Debugging — Explain how to perform debugging when something doesn't work in Cargo
- Deployment — How to deploy components to a container
  - Deployable — Deployables are archives (WAR, EAR, etc) that can be deployed in the container
  - Deployer — Performs a hot deployment of a Deployable
    - Local Deployer — Performs a hot deployment of a Deployable on a locally installed container
    - Remote Deployer — Performs a hot deployment of a Deployable on a container running on a remote machine
      - JBoss Remote Deployer
      - Jetty Remote Deployer
  - Hot Deployment — Ability to deploy/undeploy Deployables into a running container
  - JSR88 — JSR88-compliant containers support
  - Static Deployment
    - Static deployment of EAR — Deploy an EAR that will be started when the container starts
    - Static deployment of EJB — Deploy an EJB (any version) that will be started when the container starts
    - Static deployment of expanded WAR — Deploy an expanded WAR that will be started when the container starts
    - Static deployment of OSGi Bundles — Deploy an OSGi bundle that will be started when the container starts
    - Static deployment of RAR — Deploy a Java EE resource adapter (RAR) that will be started when the container starts

        - Static deployment of WAR — Deploy a WAR that will be started when the container starts
- Extensions — Extensions are additions to the Cargo core Java API such as build tool plugins, IDE plugins, etc
    - Ant support — Cargo provides Ant tasks to perform all the operations available from the Java API
    - Cargo Daemon
    - Maven2 plugin — A Maven 2 / Maven 3 plugin that wraps the Cargo Java API
        - Maven2 Archetypes
        - Maven2 Plugin Getting Started
            - Deploying to a running container
            - Generating a container configuration deployment structure
            - Merging WAR files
            - Starting and stopping a container
        - Maven2 Plugin Installation
        - Maven2 Plugin Reference Guide
        - Maven2 Plugin Tips
- Module API — API to manipulate J2EE archives, including vendor-specific deployment descriptors