This document provides an overview on the following subjects:

- Docker concepts and building blocks
- Magic xpa's usage of Docker
- Continuous delivery using Docker

Introduction

The Magic xpa platform provides code-free development and deployment. Allowing developers to leverage the same business logic to develop once and deploy across platforms, it enables the creation of a portfolio of high-performance business apps with a single skill set and minimal resources. The actual deployment of the end app can be tedious and challenging. This document provides a first glimpse at the next step taken by Magic xpa on its everlasting journey as a modern application development and deployment platform.

What is Docker?

Docker is an open platform for shipping, and running applications. Docker enables you to separate your applications from your infrastructure so you can deliver software quickly. You can significantly reduce the delay between writing code and running it in production.

More and more in the IT world, applications are being run in containers instead of virtual machines. The heart of the technology is Docker, a platform that allows users to easily pack, distribute and manage applications within containers.

The Docker adoption rate is fast, and it is rapidly becoming the de facto standard for containerization.

As an open-source project that automates the deployment of applications inside software containers, it provides fast and consistent delivery of applications, facilitates responsive deployment and scaling, and enables the running of more workloads on the same hardware. Its popularity is based on a series of important benefits, which range from fast return on investment and cost savings, standardization & productivity, compatibility & maintainability, and simplicity & faster configurations, to rapid, continuous deployment & testing, multi-cloud platforms, app & resource isolation, and security.

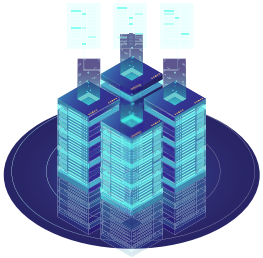
What are the Docker building blocks?

Docker images

An image is a read-only template with instructions for creating a Docker container. Often, an image is based on another image, with some additional customization. For example, you may build an image which is based on the CentOS image but installs the Apache web server and your application, as well as the configuration details needed to make your application run. You might create your own images or you might only use those created by others and published in a registry

Docker Hub

It is the world's largest repository of container images with an array of content sources including container community developers, open source projects and independent software vendors (ISV) building and distributing their code in containers



Docker Container

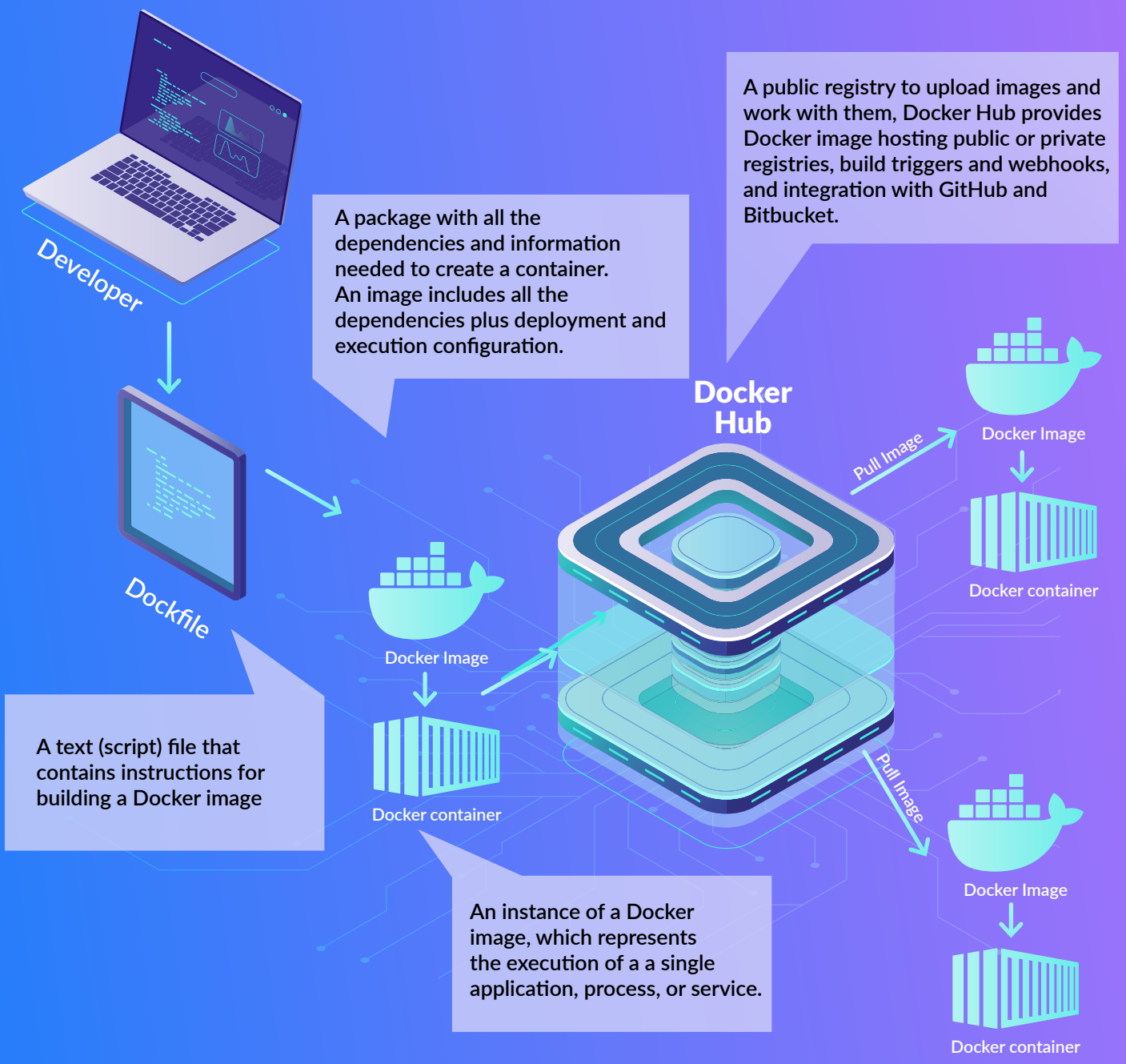
A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it. By default, a container is relatively well isolated from other containers and its host machine. You can control how isolated a container's network, storage, or other underlying subsystems are from other containers or from the host machine. A container is defined by its image as well as any configuration options you provide to it when you create or start it.

Docker Engine

An application that includes a server (a type of long-running program called a daemon process); a REST API which specifies the interfaces that programs can use to interact with the daemon; and a command line interface (CLI), as shown in the diagram.

Docker Orchestrator

Tool to package, manage, scale, and maintain containerized apps, such as Kubernetes and Docker Swarm. Development environment deployments of both of these orchestrators are provided by Docker Desktop and require very simple configuration.



How does Magic xpa on Docker fit into the continuous Delivery (CD) concept?

Docker is yet another building block in the growing world of CI-CD that Magic xpa is constantly move toward to. The availability of Magic xpa on Docker not only makes deployment easier and faster; it also provides an opportunity for Magic xpa customers to facilitate DevOps in their environment, as expected of any modern application development and deployment platform.
The constantly growing demand for continuous delivery and continuous deployment (CI-CD) means that application deployability is one of the primary concerns of software vendors today.

What different images does Magic xpa provide?

Magic xpa provides the below base images:

Application server base image: All files required for running a Magic xpa runtime engine.

Web Server base image: All files required for running a web server and the Magic xpa requester.

Broker base image: All file required for running a broker.

The below images need to be built by Devops for each specific application to be tested/deployed:

Application server image: based on the application server base image in the Docker hub, along with application specific files such as the ECF files, images and other files relevant for the proper execution of the application on the application server side.

Web Server image: based on the web server image in the Docker hub, along with any application specific files such as the Angular project files (for a web client project) relevant for the proper execution of the application on the web server side.

What are the steps for a Magic xpa developer when working with Magic xpa and Docker?

Working with both these platforms for optimal app development and deployment is a simple process, as presented in the following steps:

1. Retrieve the Docker configuration files that match a specific edition of the Magic xpa Linux server.
2. Prepare application server files and web server files that are required for the application to execute.
3. prepare application server and web server Docker images in accordance with the application specific prepared files using **Docker-compose build.**
4. Store Docker images on a Docker hub, or pass directly to the different testing and/or deployment environments.
5. Retrieve the images from Docker hub and create containers from the images, ready for testing/deployment using **docker-compose up.**

> **Note** the Docker installation package comes equipped with a clear instruction set explaining the process from A to Z.

What are the benefits of using Docker with Magic xpa?

- Encapsulation of the application components
- A secured deployment environment
- Standardization of the deployment process
- Streamlining of development and QA testing in in the different environments environments since everything runs on the same container
- Orchestration providing added value for dynamic scale aside and resource consumption based on real time resource usage.
- Containers share their operating system, so they run as isolated processes, regardless of the host operating system, on any computer, on any infrastructure and in any cloud.

Find out more about Docker and Magic xpa

Or contact us