

Docker Architecture



Какво е Docker ?

Преди да започна с архитектурата, ще ви запозная с Docker. Това е платформа за разработка, стартиране и лесно управление на приложения. Използва се на всякакви среди и операционни системи. Мога да кажа, че е най-лесния, бързия и модерен начин да си докарате приложението до продукционна среда с най-малко забавяне в доставката (deployment) на различните среди на които се разработва.

Ще започна с малко история. Преди създаването на Docker, разработчиците са използвали виртуални машини за разработката на приложенията си. Представете си че приложението ви се състои от няколко компонента. Всеки компонент се изпълнява на различна виртуална машина. Виртуалната машина на компонента има точно определена памет и процесорна сила. Обаче единия компонент поради някаква причина бива претоварен и виртуалната машина спира. Тогава трябва да се вдигне нова виртуална машина и да се стартира компонента. Това отнема много време. Друг случай е когато компонента не се натоварва изобщо и сме заделили излишно памет и процесорна сила за машината. Тогава хабим ресурсите с които разполагаме. Тук идва на помощ Docker.

Той предлага удобството да създадем контейнер, който държи отделния компонент с точно нужните ресурси. При спиране на компонента той се стартира много по-бързо от цялото стартиране на виртуална машина на ново, защото е само един процес. Докер не стартира цяла операционна система а само контейнера(изолиран процес). Цялата операция по стартирането на контейнера с приложението се извършва по-бързо по този начин. Друг плюс е когато компонента спре работа поради някакъв проблем. Тогава Docker предлага друга услуга Swarm (оркестратор на контейнерите),(като друг по-популярен начин за оркестрация на контейнери е Kubernetes), която услуга се грижи за пускането на контейнерите отново много бързо. Така подобряваме надеждността на приложението си, като не губим време за стартиране на машина и операционна система в машината. Също оркестраторите могат да пуснат още един контейнер на същия компонент ако той се претоварва от работа, но това вече е друга широка тема.

Надявам се да стана ясно с краткото обяснение.

Следващата картинка илюстрира работна среда, постигната с виртуални машини или с контейнери.

Докер използва архитектурата клиент-сървър. От модулната декомпозиция(картинката по-горе) много ясно се вижда кое е клиент и сървър. Клиента комуникира със Docker daemon(сървъра), който daemon върши най-тежката работа – изграждането, стартирането и дистрибутирането на контейнерите. Тоест всичко с контейнерите се върши от docker daemon. Клиента и daemon-а комуникират чрез REST api. Daemon-а чака заявки от клиента или от друг daemon и се грижи за изпълнението им. Това са заявки за обработка на volumes, containers, networks, images. Понеже няма подходящи думи, използвам точните наименования на английски.

Докер клиента е частта която изпраща заявките, които ние пишем към докер. Например ако напишем в command line-а на операционната система linux, на която сме инсталирали докер: “docker run ...”, клиента се грижи да изпрати правилната заявка до daemon-а. Един докер клиент може да изпраща заявки до повече от един daemon.

Докер registry е мястото където се пазят докер images. Image буквално преведено е изображение. Точно това и представлява. Изображение на някой компонент или на цяло приложение, което може да се стартира например с команда “docker run ‘image1’ ”. Като при пускането на командата, докер клиента казва на daemon-а, че той трябва да отиде в registry-то и да потърси това изображение и след това да направи контейнер от него.

Ще допълня че това обяснение на архитектурата беше когато daemon-а е на нашата операционна система, но също може да е на отдалечен сървър и така докер клиента комуникира с daemon-а от операционната система на която е клиента. А daemon-а винаги комуникира с registry-то, което играе ролята на база която е на отдалечен сървър.

Друга интересна тема е архитектурата на другата услуга на Docker – Swarm.

Как Swarm се грижи винаги контейнерите да са стартирани?

Как стартира нови контейнери за много кратко време?

Как стартира едно приложение на много еднакви контейнери с цел да намали натоварването на един контейнер?

Източници:

Информацията е базирана на официалната документация на docker: <https://docs.docker.com/>

Картинките:

<https://www.weave.works/blog/a-practical-guide-to-choosing-between-docker-containers-and-vms>

<https://morioh.com/p/47719a08c1e8>