

Давайте разберемся сначала с Docker-ом. Опять же, у Docker-а существует свое API. API – это RESTful API, то есть у нас запрос выполняется к запросу (HP3 00:14) HTTP с одним из глаголов get, post, put, delete. Ответы нам приходят, как правило, в формате Json. То есть даже на официальной страничке Docker Engine написано, что они очень стараются поддерживать API в RESTful духе, в этой парадигме. В основном у них, в принципе, получается.

Второе, о чем нужно помнить, когда мы разрабатываем API, это то, что у API существует огромное просто количество версий. Docker обновлялся довольно часто, и до сих пор обновляется довольно часто. С каждой новой версией вводятся новые изменения в API. Поэтому существует довольно-таки много референсов, описывающих различные версии API. Основы практически не меняются, меняются детали, меняются протоколы, меняются какие-то поля. Все это, в принципе, отражено в версиях API.

Docker поддерживает, соответственно, OpenAPI спецификацию. То есть вы можете ее скачать, применить к своему любимому клиенту API insomnia, postman, чем вы пользуетесь, и работать там. Вы можете работать с помощью чудесного клиента docker консольного, который умеет много, кучу всего делать, у него куча всяких различных команд, в принципе, которые отражают то, что вы можете сделать с API, и вы будете скорее всего пользоваться в повседневной жизни. И в принципе, все эти же команды реализованы в SDK. Давайте посмотрим на него подробнее.

Итак, наш Docker (HP3 01:56), наш Docker daemon, который, непосредственно, отвечает за все эти взаимодействия с Docker-ом, за запуск контейнеров и за их взаимодействие друг с другом, за саму контейнеризацию. Также отвечает за принятие и обработку запросов по HTTP. Соответственно, способов обработать запрос у нас три, способов поднять, так называемый, Docker API. Первое – это системные сокеты на unix. Это дефолтный режим для Docker-а, когда он

поднимает сокет, и вы в него можете писать, из него читать. Поскольку это самый быстрый и безопасный способ, он является умолчальным.

Как правило, Docker у нас работает на локальной машине. Поэтому, почему бы и не да. Соответственно, в случаях, когда у вас Docker лежит на другой точке в системе, вы можете поднять Docker через TCP, и он по умолчанию поднимается незащищенным, то есть поверх него нужно положить, соответственно, HTTPS. Вы можете использовать встроенный HTTPS Incriptions, чтобы подключаться по SSL. И порт по умолчанию для Docker-а – является порт 2375. Опять же, это все в деталях знать не надо до тех моментов, пока вы не занимаетесь обслуживанием сети Docker-ов. Если вы этим занимаетесь, вы это знаете.

Итак, для наших изысков и экспериментов я буду использовать поднятый Docker for desktop. Он существует для MAC, для Windows, для Linux. Соответственно, он очень удобен чем? Тем, что..., много чего, что доступно только в консольном приложении Docker, здесь дублируется – опять же, список image-ей, список, так называемых, Volume-ов. То есть подсоединенных дисков контейнеров, список запущенных контейнеров. Опять же, здесь существуют всякие различные настройки. Можно сразу поднять, без шифрования, значит, наш Docker на tcp://localhost:2375. Для Windows тут я могу еще WSL based engine включить совместимость, соответственно, с WSL-ем. Очень полезная штука. Рекомендую установить перед тем, как мы начнем уже работать, непосредственно, с API.