|  |  |  |
| --- | --- | --- |
|  | **Команда** | **Описание** |
| **1** | **docker help** | Cправка по docker |
| **1** | **docker [command]** --**help** | Справка по конкретной команде.  Пример:  *docker container --help* |
| **1** | **docker system info** | Просмотр версии docker |
| **2** | **docker image pull hello-world** | Скачивание образа (image) с названием **hello-world**  с ресурса DockerHub (https://hub.docker.com/) |
| **2** | **docker container create --name first *hello-world*** | Создание из образа ***hello-world*** контейнер с именем **first** |
| **2** | **docker container start first или docker container start** <id> | Запуск контейнер с именем **first** или по id (пример: b4aa96) |
| **3** | **docker container run --name first *hello-world*** | Запуск контейнера с именем **first** из образа ***hello-world*** (если на компьютере нет образа - он скачается из репозитория) |
| **3** | **docker container ls --all** | Просмотр всех (работающие и неработающие) существующих контейнеров |
| **3** | **docker container ls** | Просмотр только работающих в данный момент контейнеров |
| 4 | docker container run -d  ***-e*** MYSQL\_RANDOM\_ROOT\_PASSWORD=yes ***-e*** MYSQL\_DATABASE=app ***-e*** MYSQL\_USER=app ***-e*** MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef  mysql | Запуск контейнер mysql в автономном режиме (***-***d сокращение от detach) и передача внутрь контейнера переменных окружения (***-e***) MYSQL\_RANDOM\_ROOT\_PASSWORD, MYSQL\_DATABASE… Автономный режим нужен когда контейнер запустился и работает постоянно. Если запустить без автономного режима он займет консоль. В автономном режиме мы сможем дальше работать в консоли. |
| 4 | docker container stop <id> | Остановка контейнер с <id> |
| 5 | docker container run -d \    -e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes \    -e MYSQL\_DATABASE=app \    -e MYSQL\_USER=app \    -e MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef \    -p 3306:3306 \    mysql | Флаг -p определяет binding портов (привязка порта контейнера к порту хоста).  Порт 3306 контейнера привязывается к порту 3306 хоста.  Пример:  внутри контейнера на порту 3306 запущена база данных mysql. Чтобы подключиться к ней с нашего ПК нужно пробросить порт из контейнера наружу. После выполнения команды на нашем ПК на порту 3000 мы получим доступ к базе (приложению, работающее в контейнере на порту 3306). В документации хороших образов мы можем посмотреть какие порты биндить docker container ls  и смотрим в колонку PORTS |
| 6 | docker container ***exec*** -it <id> bash | В запущенном контейнере <id> выполняется команда (***exec***) открытия терминала (bash) в интерактивном режиме (-it). Интерактивный режим нужен чтобы в режиме реального времени взаимодействовать с  потоком вывода контейнера |
| 7 | docker container run -d \    -e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes \    -e MYSQL\_DATABASE=app \    -e MYSQL\_USER=app \    -e MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef \    -p 3306:3306 \    -v "$PWD/data":/var/lib/mysql \    mysql | Из папки внутри контейнера /var/lib/mysql   копирует все данные в папку $PWD/data $PWD — это текущий каталог в Linux. В Windows нужно использовать %cd% для CMD. |
|  | docker-compose up | Запускает все сервисы, прописанные в файле docker-compose.yml Запускать команду нужно из папки в которой лежит файл docker-compose.yml |
|  | docker-compose down | Остановит и удалит все  контейнеры, которые запущены командой  docker-compose up |
|  | docker node ls |  |
|  | docker exec -it --user [username] [container] bash | выполнить внутри контейнера команду от имени пользователя [username] |
| **Yandex Cloud commands** | | |
|  | yc init | залогиниться |
|  | yc vpc network create \      --name net \      --labels my-label=netology \      --description "my first network via yc" | Создать сеть с именем net в папке netology |
|  | yc vpc subnet create \      --description "my first subnet via yc"\      --name my-subnet-a \      --network-name net \      --range 10.1.2.0/24 \      --zone ru-central1-a | создать подсеть |
|  | yc compute image list | показать имеющиеся образы |
|  | yc iam key create --service-account-name service-account -o key.json | создать ключ для сервисного аккаунта |
|  | yc vpc subnet delete --name my-subnet-a |  |
|  | yc vpc network delete --name net |  |
|  | yc compute image delete --id <id> | удалить созданный ранее образ ОС |
| **Packer** | | |
|  | packer validate <file.json> |  |
|  | packer build <file.json> |  |
| **Terraform** | | |
|  | terraform init |  |
|  | terraform validate |  |
|  | terraform plan |  |
|  | terraform apply -auto-approve |  |
|  | terraform destroy -auto-approve | удалим созданные ВМ + подсеть + сеть в YA Cloud если они были созданы |
| **Ansible** | | |
|  | ansible-playbook provision.yaml | Запуск плэйбука |

docker container run -d ***-e*** MYSQL\_RANDOM\_ROOT\_PASSWORD=yes ***-e*** MYSQL\_DATABASE=app ***-e*** MYSQL\_USER=app ***-e*** MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef mysql

docker container run -d -e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes -e MYSQL\_DATABASE=app -e MYSQL\_USER=app -e MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef -p 3306:3306 mysql

C:\Users\msnos\Desktop>docker container run -d -e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes -e MYSQL\_DATABASE=app -e MYSQL\_USER=app -e MYSQL\_PASSWORD=9mREsvXDs9Gk89Ef -p 3306:3306 -v "%cd%/data":/var/lib/mysql mysql