Link to Tutorial: <https://www.docker.com/101-tutorial/>

Link to zip file: [Download the ZIP](http://localhost/assets/app.zip). Open the ZIP file and make sure you extract the contents.

Important commands:

|  |  |
| --- | --- |
| **Topic** | **Command** |
| **Getting Started:** |  |
| Starting the tutorial | docker run -d -p 80:80 docker/getting-started |
| **Our Application:** |  |
| Building the container image: | docker build -t getting-started . |
| Starting your container using the docker run command and specify the name of the image | docker run -dp 3000:3000 getting-started |
| **Updating our App:** |  |
| Get the ID of the container | docker ps |
| To stop the container | # Swap out <the-container-id> with the ID from docker ps  docker stop <the-container-id> |
| To Remove the container | docker rm <the-container-id> |
| You can stop and remove a container in a single command | docker rm -f <the-container-id> |
| **Sharing our App:** |  |
| To give the getting-started image a new name | docker tag getting-started YOUR-USER-NAME/getting-started |
| Pushing your image to your repo/Docker Hub | docker push YOUR-USER-NAME/getting-started |
| **Persisting our DB:** |  |
| Create a volume | docker volume create todo-db |
| Start the todo app container, but add the -v flag to specify a volume mount, to capture all files created at the path. | docker run -dp 3000:3000 -v todo-db:/etc/todos getting-started |
| To see where Docker is storing your data | docker volume inspect todo-db |
| **Using Bind Mounts:** |  |
| To run your container to support a development workflow | docker run -dp 3000:3000 `  -w /app -v "$(pwd):/app" `  node:12-alpine `  sh -c "yarn install && yarn run dev" |
| To watch the logs | docker logs -f <container-id> |
| **Starting MySQL:** |  |
| Create the network | docker network create todo-app |
| Start a MySQL container and attach it to the network. | docker run -d `  --network todo-app --network-alias mysql `  -v todo-mysql-data:/var/lib/mysql `  -e MYSQL\_ROOT\_PASSWORD=secret `  -e MYSQL\_DATABASE=todos `  mysql:5.7 |
| Connect to the database and verify it connects | docker exec -it <mysql-container-id> mysql -p |
| List the databases and verify you see the todos database | mysql> SHOW DATABASES;  or  mysql> SHOW SCHEMAS; |
| Start a new container using the nicolaka/netshoot image | docker run -it --network todo-app nicolaka/netshoot |
| To look up the IP address for the hostname mysql | dig mysql |
| To specify each of the environment variables, as well as connect the container to the app network | docker run -dp 3000:3000 `  -w /app -v "$(pwd):/app" `  --network todo-app `  -e MYSQL\_HOST=mysql `  -e MYSQL\_USER=root `  -e MYSQL\_PASSWORD=secret `  -e MYSQL\_DB=todos `  node:12-alpine `  sh -c "yarn install && yarn run dev" |
| To look at the logs for the container | docker logs <container-id> |
| Connect to the mysql database and prove that the items are being written to the database. | docker exec -it <mysql-container-id> mysql -p todos |
| **Using Docker Compose:** |  |
| To see version of Docker Compose | docker-compose version |
| Start up the application stack | docker-compose up -d |
| look at the logs | docker-compose logs -f |
| **Image Building Best Practices:** |  |
| To scan the getting-started image | docker scan getting-started |
| To see the layers in the getting-started image | docker image history getting-started |