

## Commands to manager Docker Container

Run Command	<code>docker run {image-name}</code>
Override startup command	<code>docker run {image-name} {default-command-override}</code>
List all running containers	<code>docker ps</code>
List all containers ever created on this machine	<code>docker ps --all</code>
Create a container	<code>docker create {image-name}</code>
Start a container	<code>docker start {container-id}</code>
Start a container with attaching output	<code>docker start -a {container-id}</code>
Destroy all containers and image cache	<code>docker system prune</code>
Check logs of a container	<code>docker logs {container-id}</code>
Stop a container gracefully	<code>docker stop {container-id}</code>
Kill a container	<code>docker kill {container-id}</code>
Execute additional commands in a container	<code>docker exec -it {container-id} {command}</code>
Get Shell access in a container	<code>docker exec -it {container-id} sh</code>
Get Shell access at the start of the container	<code>docker run -it {image-name} sh</code>
Run a container with port mapping	<code>docker run -p {outer-port} : {container-port} {image-id}</code>
Run a container in detached mode	<code>docker run -d {image-id}</code>
Run a container with Volume Mapping	<code>docker run -v {outer-folder}:{container-folder} {image-id}</code>
Run a container with Volume Bookmark	<code>docker run -v {container-folder} {image-id}</code>

## Commands to Build Docker Image

Build Command	<code>docker build .</code>
Build Command with an Image tag	<code>docker build -t {tag-name} .</code>
Build Command with renamed Docker file	<code>docker build -f {new-docker-file} .</code>
Create an image from a container	<code>docker commit -c 'CMD ["default-start-up-command"]' {container-id}</code>

## Dockerfile Commands

Specify Base Image	FROM alpine
Install packages in Alpine	RUN apk add --update redis
Specify default startup command	CMD ["redis-server"]
Copy source directory files inside the container	COPY {source-dir} {destination-dir}
Change the working directory	WORKDIR {working-dir-folder}

Dockerfile - Alpine Basic

```
FROM alpine
RUN apk add --update redis
CMD ["redis-server"]
```

Dockerfile - Node App

```
FROM node:alpine
WORKDIR /usr/app
COPY ./ ./
RUN npm install
CMD ["npm", "start"]
```

Dockerfile - Node App with Build Optimization

```
FROM node:lpine
WORKDIR /usr/app
COPY ./package.json ./
RUN npm install
COPY ./ ./
CMD ["npm", "start"]
```

Dockerfile - Multi-step build process

```
# Build Phase
FROM node:alpine as builder
WORKDIR '/app'
COPY ./package.json ./
RUN npm install
COPY ./ ./
RUN npm run build

# Run Phase
FROM nginx
COPY --from=builder /app/build /usr/share/nginx/html
```

Docker Compose Commands

Run an image	docker-compose up
Build an image	docker-compose build
Build and run an image	docker-compose --build
Run an image in the background	docker-compose up -d
Stop running containers	docker-compose down
Check container status	docker-compose ps

Docker Compose Basic YML file

```
version: "3"
services:
  redis-server:
    image: 'redis'
  node-app
    build: .
    ports:
      - "4001:8081"
```

Docker Compose YML file with Restart Policy

```
version: "3"
services:
  redis-server:
    image: 'redis'
  node-app
    restart: always
    build: .
    ports:
      - "4001:8081"
```

Docker Compose YML file with Volume Mapping & Volume Bookmark

```
version: "3"
services:
  web:
    build: .
    ports:
      - 3000:3000
    volumes:
      - /app/nod_modules
      - ./app
```

Docker Compose YML file with Renamed Docker file

```
version: "3"
services:
  web:
    build:
      context: .
      dockerfile: Dockerfile.dev
    ports:
      - 3000:3000
    volumes:
      - /app/nod_modules
      - ./app
```

Git Commands

Create a local Repo	git init
Add files in the staging area	git add .
Commit files	git commit -m {commit-message}
Map to Github remote repo	git remote add origin {github-repo-url}
Push changes to Github	git push origin master

Travis CI “YML” file (.travis.yml) for test set up

```
sudo: required
services:
  docker
before_install:
  docker build -t stephengrider/docker-react -f Dockerfile.dev .
scripts:
```

```
docker run stephengrider/docker-react npm run test -- --coverage
```

## Travis CI “YML” file (.travis.yml) for test and deployment set up

```
sudo: required
services:
  docker
before_install:
  docker build -t stephengrider/docker-react -f Dockerfile.dev .
scripts:
  docker run stephengrider/docker-react npm run test -- --coverage
deploy:
  provider: elasticbeanstalk
  region: "us-west-1"
  app: "docker"
  env: "Docker-env"
  bucket-name: "elasticbeanstalk-us-west-2-38483583445"
  bucket-path: "docker"
  on:
    branch: master
```