

Docker Meetup HK
Jul 2017

Demo Steps

Demo1

- Introducing docker playground (play-with-docker.com)
- Show version
 - docker version
- Start a simple nginx container
 - docker run -d -p 8080:80 nginx
 - Click link to visit the home page of nginx
- Create a simple static file (html/index.html)
- Run nginx again with volume
 - docker run -d -p 8080:80 -v /root/html:/usr/share/nginx/html nginx
 - Show the page content with the mounted volume
- Install play-with-docker Chrome extension
- Go to hub.docker.com
- Find redis
- Click “Try in PWD”
- Show the redis redmon interface
- Play with some simple commands

Demo 2

- Introduce the demo application
 - Refer to github page
- docker-contact-db
 - Discuss the Dockerfile instructions
 - Initdb.sql
 - Build/tag/push/run the image
- Docker-contact-api
 - Discuss the application (app.ts)
 - Start the app, show it's connected to DB
 - Run curl to show the sample response
 - Discuss the Dockerfile
 - build/tag/push the image
- Docker-contact-web
 - Discuss the application (Angular/Material/TypeScript)
 - Build the application (ng build)
 - Build/tag/push
- Run the contact app locally
 - Discuss the docker-compose.yml file
 - Run the app and show it
- Create a Swarm cluster
 - Visit play with docker
 - Init a swarm
 - `docker swarm init --advertise-addr 10.0.16.3`
 - Create a new instance and join the swarm
- Discuss the docker compose file for swarm deployment
 - docker-compose-swarm.yml
- Create secret for postgresSQL password
 - Create a file passwd.txt
 - `docker secret create contact_db_passwd passwd.txt`
 - `docker secret inspect [id]`
- Clone docker-contact repository
 - git clone <https://github.com/clarenceh/docker-contact.git>
- Start the application
 - cd to docker-contact-app folder
 - `docker stack deploy --compose-file docker-compose-swarm.yml contact`
- Show the application
 - Go to db container and show the tables
 - Show the logs of service docker-contact-api
 - Show the endpoint (9001) that api is working
 - Show the secret that was injected into the DB and nodejs containers
 - Show the frontend and not working because the endpoint was wrong
- Update the frontend
 - Modify the environment.prod.ts file for correct endpoint

- ng build --prod
 - Build/tag/push the image
- Update the contact app on swarm cluster
 - Update the docker-compose-swarm.yml to use frontend 1.0.0-prd
 - docker stack deploy --compose-file docker-compose-swarm.yml contact
- Scale up the api service
 - docker service scale contact_api=2
- Show the api endpoint is doing load balancing
 - curl -v http://pwd10-0-16-3-9001.host1.labs.play-with-docker.com/contacts | jq
- Show the consolidated service logs of api service
- Show the frontend is also using load balancing
- Run Docker swarm visualizer
 - docker run -it -d -p 9500:8080 -v /var/run/docker.sock:/var/run/docker.sock clarenceb/visualizer
 - Show the visualizer frontend and explain to audience