Docker : Docker MasterClass for DevOps

Lab : Deploy Multi-Node Application in Docker Swarm

Docker : Service Containers

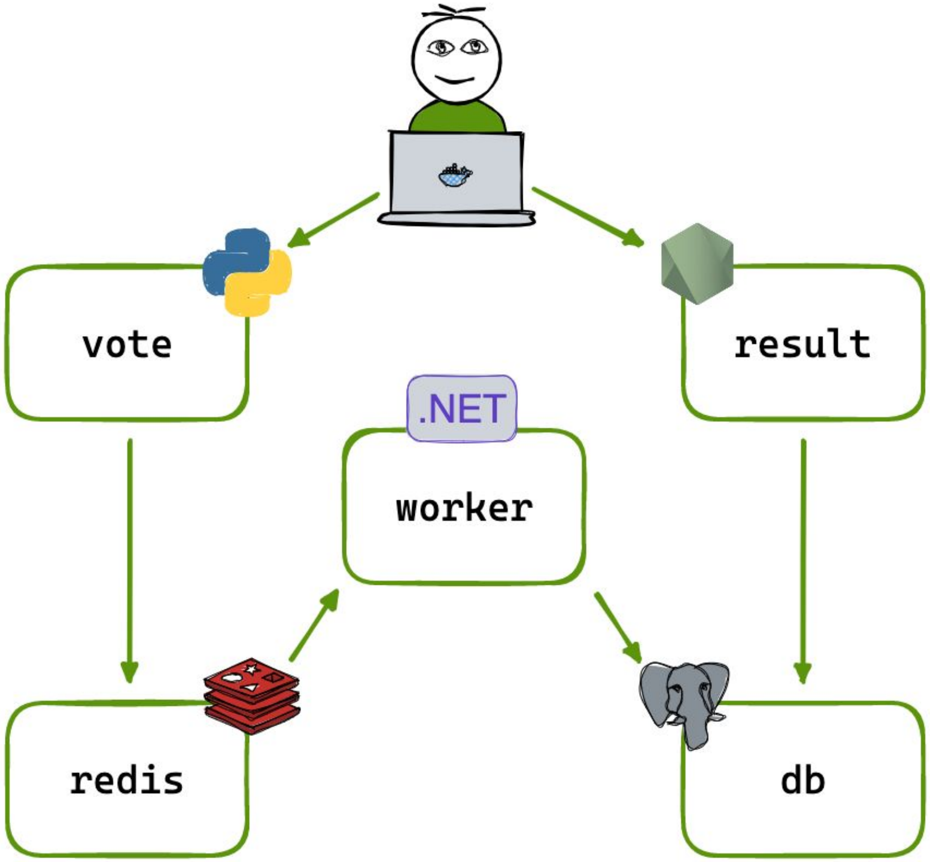
➢ Assignment to Deploy Multi Node Service via Docker Swarm

➢ Docker’s Distributed Voting App.

➢ This solution uses Python, Node.js, .NET, with Redis for messaging and Postgres for storage.

Docker : Service Containers

- ➢ Architecture –



Docker : Service Containers

- This is combination of Several docker and compose/swarm file. Code can view at below location: https://github.com/dockersamples/example-voting-app

- App is designed by Docker community and available on Docker Hub for Public use.

- This service need **1 Mount Volume, 2 Network and 5 Stack Services**

- Two overlay network you can call **frontend** and **backend** is needed.

```
docker network create -d overlay frontend_ntw
docker network create -d overlay backend_ntw
```

Docker : Service Containers

- Voting App:
  - Image : dockersamples/examplevotingapp_vote
  - Web front app
  - Publish this on port **5000**, Listener Container Port **80**
  - Publish **4 replicas**
  - Publish on **frontend** overlay Network

```
docker service create --name vote -p 5000:80 --network frontend_ntw
--replicas 4 dockersamples/ examplevotingapp_vote
```

Docker : Service Containers

- Redis:
    - Image : redis:alpine
    - Redis is used to Collect New Votes
    - Publish **4 replicas**
    - Publish on **frontend** overlay Network

```
docker service create --name redis --network frontend_ntw --replicas 4 redis:alpine
```

Docker : Service Containers

- Worker:
  - Image : dockersamples/examplevotingapp_worker
  - A .NET worker which consumes votes and stores them
  - Publish **4 replicas**
  - Publish on **frontend** & **backend** overlay Network

```
docker service create --name worker --network fronend_ntw --network
backend_ntw --replicas 4 dockersamples/examplevotingapp_worker
```

Docker : Service Containers

- DB Service:
  - Image : postgres:15-alpine
  - Mount Volume and mount to /var/lib/postgresql/data
  - Publish on backend network
  - Publish 1 replicas

```
docker service create --name db --network backend_ntw --mount
type=volume,source=db-data,target=/var/lib/postgresql/data -e
POSTGRES_PASSWORD=mypass postgres:15-alpine
```

Docker : Service Containers

- Result Service:
  - Image : dockersamples/examplevotingapp_result
  - Will display the Voting result
  - Publish on **backend** network
  - Publish on port **5001**, Container port **80**
  - Publish 1 replicas

```
docker service create --name result --network backend_ntw -p 5001:80
dockersamples/examplevotingapp_result
```

Thank You...

Don’t be the Same! Be Better!!!