Docker Networks

When using Docker Compose we can specify additional networks for segmentation. This allows us to have containers running on different networks depending on security needs.

This compose file builds the application from source code, and then creates two networks, and a couple volumes.

Networks

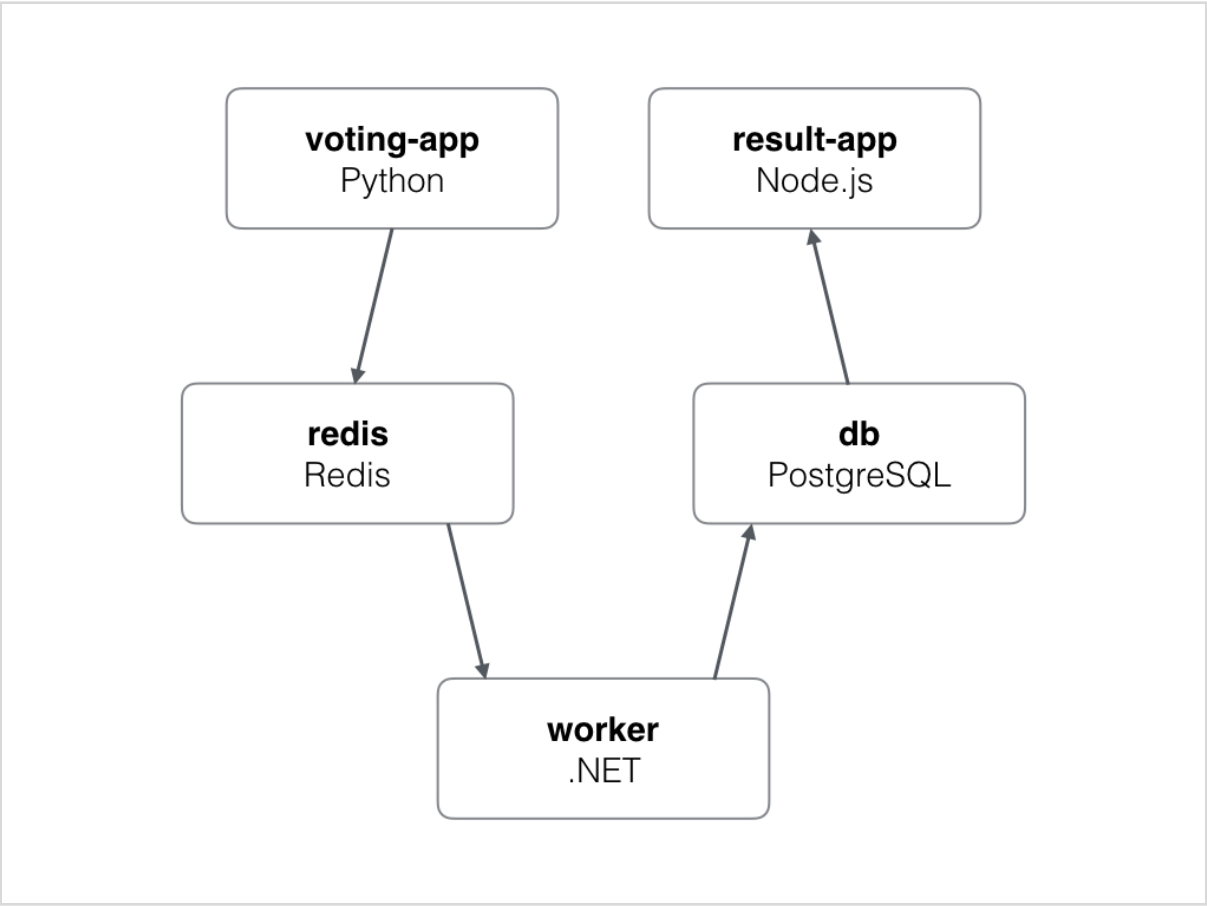
- front-tier
- back-tier

Review the example compose file `compose-network/docker-compose.yml`

```
version: "3"

services:
  vote:
    build: ./vote
    command: python app.py
    volumes:
     - ./vote:/app
    ports:
      - "5000:80"
    networks:
      - front-tier
      - back-tier

  result:
    build: ./result
    command: nodemon server.js
    volumes:
      - ./result:/app
    ports:
      - "5001:80"
      - "5858:5858"
    networks:
```

```
      - front-tier
      - back-tier

  worker:
    build:
      context: ./worker
    depends_on:
      - "redis"
    networks:
      - back-tier

  redis:
    image: redis:alpine
    container_name: redis
    ports: ["6379"]
    networks:
      - back-tier

  db:
    image: postgres:9.4
    container_name: db
    volumes:
      - "db-data:/var/lib/postgresql/data"
    networks:
      - back-tier

volumes:
  db-data:

networks:
  front-tier:
  back-tier:
```

The architecture of the application is:



Now let's launch the application

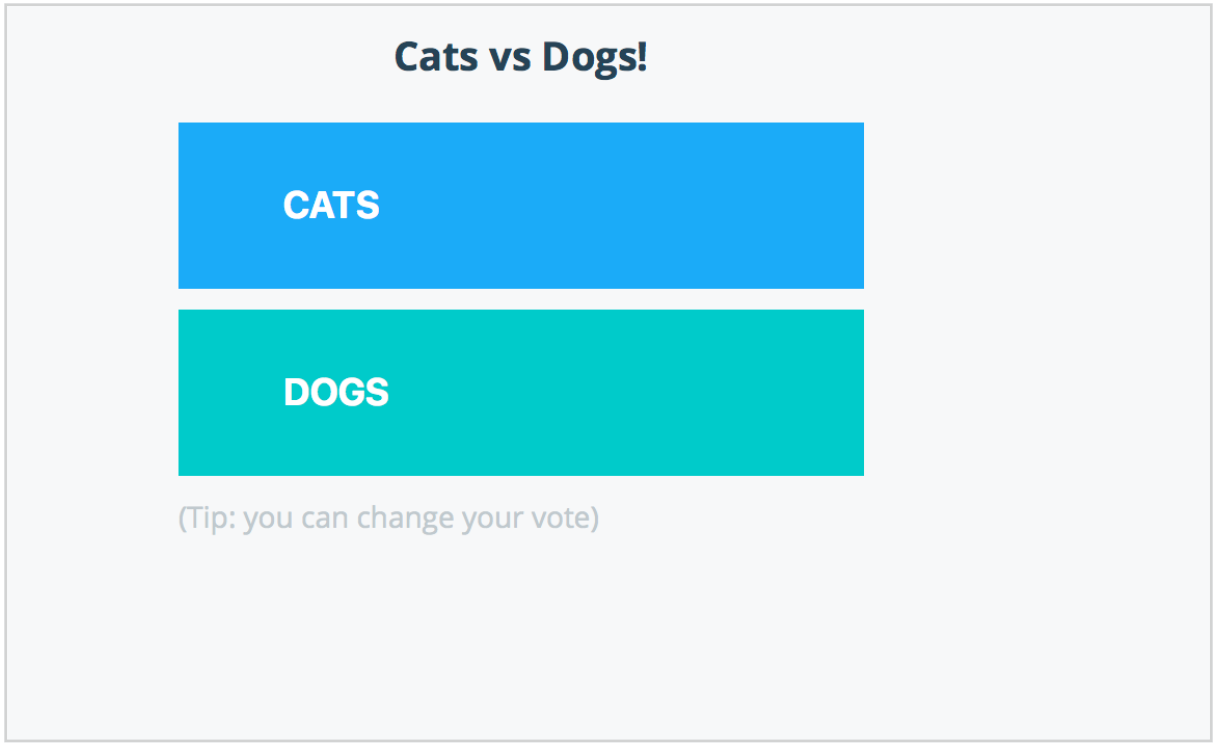
```
docker-compose -f compose-network/docker-compose.yml up -d
```

Confirm the application started successfully

```
docker-compose ps
```

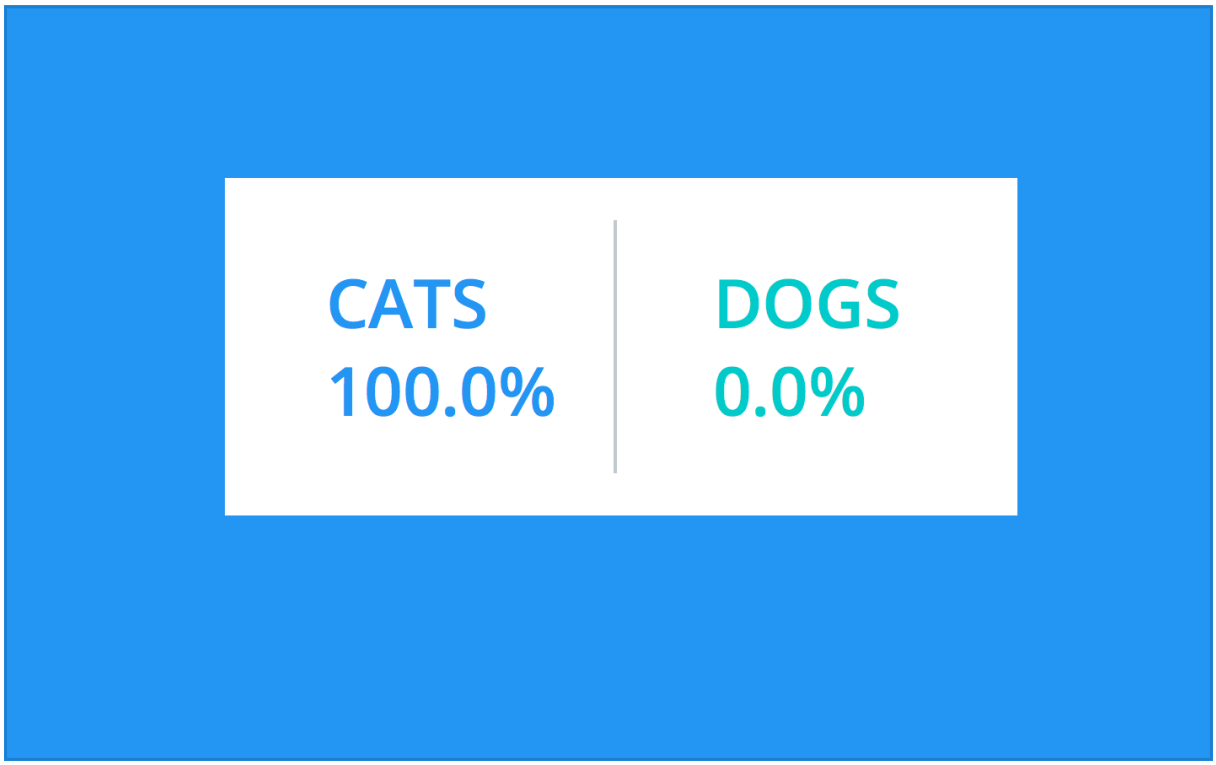
Now if things look good load up the voting app and the results app in a browser.

Voting Application
http://localhost:5000/



Results Application

http://localhost:5001/



Lab Complete