

lab03

docker-compose.yml

```
version: "3"
services:
  registry:
    container_name: registry
    image: registry:latest
    ports:
      - 5000:5000
    volumes:
      - "./registry-data:/var/lib/registry"

  manager:
    container_name: manager
    image: docker:19.03.5-dind
    privileged: true
    tty: true
    ports:
      - 8080:80
      - 9000:9000
    depends_on:
      - registry
    expose:
      - 3375
    command: "--insecure-registry registry:5000"
    volumes:
      - "./stack:/stack"

  worker01:
    container_name: worker01
    hostname: worker01
    image: docker:19.03.5-dind
    privileged: true
    tty: true
    ports:
      - 8000:8000
    depends_on:
      - manager
      - registry
    expose:
      - 7946
      - 7946/udp
      - 4789/udp
    command: "--insecure-registry registry:5000"

  worker02:
    container_name: worker02
    hostname: worker02
    image: docker:19.03.5-dind
    privileged: true
    tty: true
```

```

ports:
  - 5001:5001
depends_on:
  - manager
  - registry
expose:
  - 7946
  - 7946/udp
  - 4789/udp
command: "--insecure-registry registry:5000"

```

```

worker03:
  container_name: worker03
  image: docker:19.03.5-dind
  privileged: true
  tty: true
  depends_on:
    - manager
    - registry
  expose:
    - 7946
    - 7946/udp
    - 4789/udp
  command: "--insecure-registry registry:5000"

```

todo-app.yml

```

version: "3"
networks:
  frontend:
    driver: overlay
    external: true
  backend:
    driver: overlay
    external: true
services:
  db:
    image: postgres:9.4
    deploy:
      replicas: 1
      placement:
        constraints: [node.role != manager]
    networks:
      - backend
    volumes:
      - db-data:/var/lib/postgresql/data
  result-app:
    image: bretfisher/examplevotingapp_result
    deploy:
      replicas: 1
      placement:
        constraints: [node.hostname == worker02]
    ports:
      - 5001:80
    depends_on:
      - db

```

```

    networks:
      - backend

redis:
  image: redis:3.2
  deploy:
    replicas: 1
    placement:
      constraints: [node.role != manager]
  networks:
    - frontend
worker:
  image: bretfisher/examplevotingapp_worker:java
  deploy:
    replicas: 1
    placement:
      constraints: [node.role != manager]
  depends_on:
    - redis
  networks:
    - frontend
    - backend
voting-app:
  image: bretfisher/examplevotingapp_vote
  ports:
    - 8000:80
  deploy:
    replicas: 3
    placement:
      constraints: [node.hostname == worker01]
  depends_on:
    - worker
  networks:
    - frontend

volumes:
  db-data:

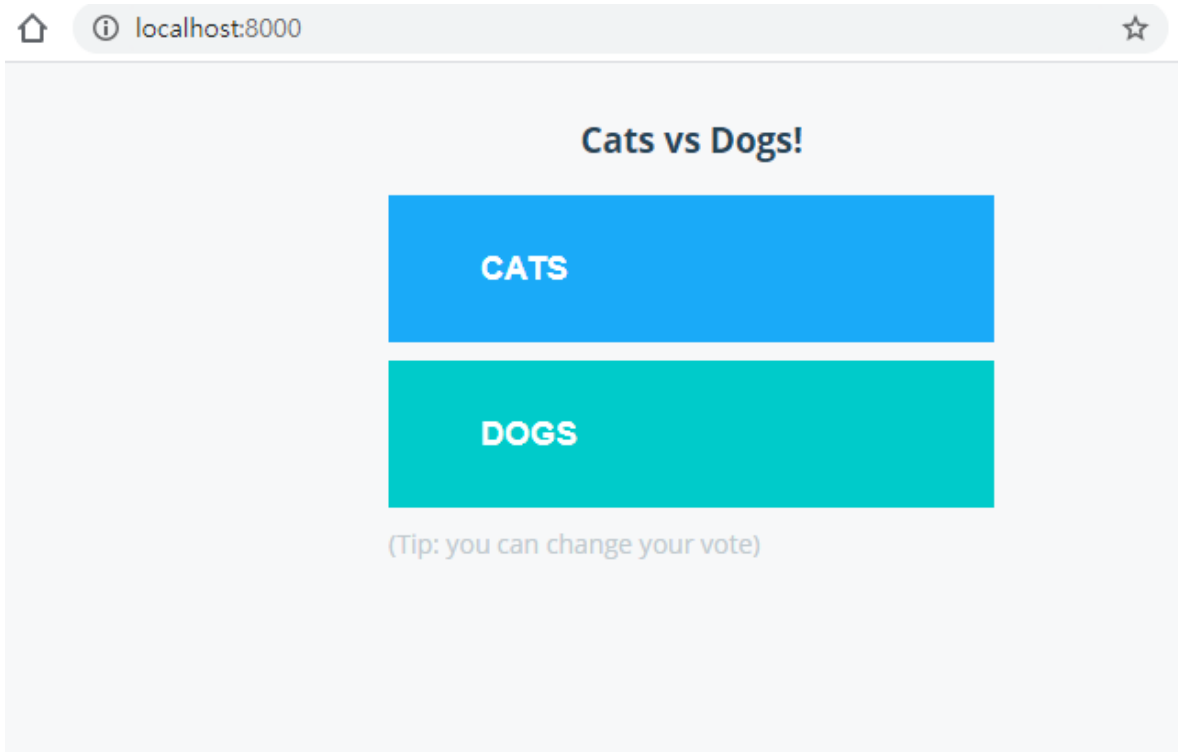
```

bash

```

PS C:\Users\HPE\work\docker\lab\2.swarm\swarm-app-1> docker-compose up
# docker exec -it manager sh
# docker exec -it manager docker swarm init
# docker exec -it worker01 docker swarm join ...
# docker exec -it worker02 docker swarm join ...
# docker exec -it worker03 docker swarm join ...
# docker exec -it manager docker network create --driver=overlay --attachable
frontend
# docker exec -it manager docker network create --driver=overlay --attachable
backend
# docker exec -it manager docker stack deploy -c /stack/todo-app.yml my-vote-app
# docker service ls

```



CATS 를 누르면

