docker birthday - Wiki - # 12

Workshop

Create a Docker Swarm with:

- a mysql container
- a redmine container
- several nginx containers

Share Data with Convoy Volume Driver (over NFS)

Docker Swarm

- create Docker Swarm (swarm mode)

```
 docker init swarm
```

```
docker swarm join ...
```

```
docker node ls
```

registry

/etc/docker/daemon.json:

```
{
    "bridge": "docker0",
    "debug": true,
    "icc": false,
    "insecure-registries": ["birthday1:5000"],
    "ip": "0.0.0.0",
    "iptables": true,
    "ipv6": false,
    "ip-forward": true,
    "ip-masq": true,
    "live-restore": false,
    "log-driver": "",
    "log-level": "",
    "log-opts": {},
    "max-concurrent-downloads": 3,
    "max-concurrent-uploads": 5,
    "mtu": 0,
    "oom-score-adjust": -500,
    "raw-logs": false,
    "selinux-enabled": false,
    "storage-driver": "aufs",
    "userland-proxy": false,
    "userns-remap": ""
}
```

Dockerfile:

```
FROM registry:2

RUN apk update && apk upgrade && apk add curl
```

docker-compose.yml:

```
  registry:
    restart: always
    build: registry
    #container_name: domo_registry
    ports:
    - 5000:5000
    volumes:
      - registry:/var/lib/registry
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost:5000/ || exit 1"]
      interval: 1m
      timeout: 10s
      retries: 3
```

nginx

Dockerfile:

```
FROM nginx:1.11.8-alpine

RUN apk update && apk upgrade && apk add curl

COPY nginx.conf /etc/nginx/nginx.conf
```

nginx.conf:

```
#user  nobody;
worker_processes  1;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    #tcp_nopush     on;
    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server {
      ### server port and name ###
      listen          80;

      ### log files ###
      access_log  /var/log/nginx/access.log combined;
      error_log  /var/log/nginx/error.log notice;

      index index.php index.html index.htm;

      error_page 404 /404.html;

      error_page 500 502 503 504 /50x.html;

      location ~^/(404|50x).html$ {
        root /usr/share/nginx/html;
      }

      root /usr/src/redmine/public;

      location ^~ /(images|themes|stylesheets|javascript)/ {
        gzip_static on;
        expires max;
        add_header Cache-Control public;
      }

      try_files $uri/index.html $uri @redmine;
      location @redmine {
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;
        proxy_redirect off;
        proxy_pass http://bday_redmine:3000;
      }

      keepalive_timeout 10;
    }
}
```

bday.yml:

```
 nginx:
    # build nginx with Dockerfile in ./nginx
    # see https://hub.docker.com/_/nginx/
    image: birthday1:5000/nginx
    # map localhost:8080 to nginx:80
    ports:
      - 80:80
    depends_on:
      - redmine
    # access public volume from redmine
    volumes:
      - redmine_public:/usr/src/redmine/public
    networks:
      - backend
    depends_on:
      - redmine
    deploy:
      mode: global
      #replicas: 3
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost/ || exit 1"]
      interval: 1m
      timeout: 10s
      retries: 3
```

redmine

Dockerfile:

```
FROM redmine:latest

RUN apt-get update && apt-get upgrade -y -o DPkg::Options::=--force-confold && \
  DEBIAN_FRONTEND=noninteractive apt-get install --no-install-recommends -y -o DPkg::Options::=--f
orce-confold \
  curl \
  && apt-get clean

COPY pepper /usr/src/redmine/public/themes/pepper
```

bday.yml:

```
volumes:
  # persistent volume for redmine uploads
  redmine_files:
    driver: convoy
  # persistent volume for redmine public
  redmine_public:
    driver: convoy

services:
  redmine:
    # use redmine:latest image from docker hub
    image: birthday1:5000/redmine
    build: redmine
    # setup redmine environment
    # see https://hub.docker.com/_/redmine/
    environment:
      - REDMINE_DB_MYSQL=bday_db
      - REDMINE_DB_USERNAME=redmine
      - REDMINE_DB_PASSWORD=r3dmin3
    depends_on:
      - db
    # use persinstent volume for file uploads
    # make public folder accessable for nginx
    volumes:
      - redmine_files:/usr/src/redmine/files
      - redmine_public:/usr/src/redmine/public
    networks:
      - backend
    deploy:
      replicas: 1
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost:3000/ || exit 1"]
      interval: 1m
      timeout: 10s
      retries: 3
```

mysql

Dockerfile:

```
FROM mysql:5
COPY redmine.dmp.sql /docker-entrypoint-initdb.d/
# modified docker-entrypoint.sh with CREATE USER '$MYSQL_USER'@'%.%'
COPY docker-entrypoint.sh /usr/local/bin/docker-entrypoint.sh
```

bday.yml:

```
volumes:
  # persistent volume for mysql databases
  mysql_data:

  db:
    # use mysql 5 image from docker hub
    image: birthday_mysql
    build: mysql
    # setup mysql environment
    # see https://hub.docker.com/_/mysql/
    environment:
      - MYSQL_ROOT_PASSWORD=s3cr3t
      - MYSQL_DATABASE=redmine
      - MYSQL_USER=redmine
      - MYSQL_PASSWORD=r3dmin3
    networks:
      - backend
    # use persinstent volume for DBs
    volumes:
      - mysql_data:/var/lib/mysql
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.role == manager
          - node.hostname == birthday1
    stop_grace_period: 30s
```

Deployment

```
docker stack deploy -c bday.yml bday
```

Stretch goals

Convoy Volume Driver

/etc/fstab:

```
birthday-nfs:/var/opt/nfs /var/opt/nfs nfs4 rsize=8192,wsize=8192,timeo=14,intr    0       0
```

Install convoy:

```
wget https://github.com/rancher/convoy/releases/download/v0.5.0/convoy.tar.gz
tar xvzf convoy.tar.gz
sudo cp convoy/convoy convoy/convoy-pdata_tools /usr/local/bin/
sudo mkdir -p /etc/docker/plugins/
sudo bash -c 'echo "unix:///var/run/convoy/convoy.sock" > /etc/docker/plugins/convoy.spec'
```

/etc/systemd/system/docker-convoy.service:

```
[Unit]
Description=Docker Convoy Volume Plugin
Before=docker.service
After=network.target
Requires=docker.service

[Service]
ExecStart=/usr/local/bin/convoy daemon --drivers vfs --driver-opts vfs.path=/var/opt/nfs
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

Start convoy volume daemon:

```
systemctl enable docker-convoy.service 
systemctl start docker-convoy.service
```

redmine.yml:

```
volumes:
  # persistent volume for redmine uploads
  redmine_files:
    driver: convoy
  # persistent volume for redmine public
  redmine_public:
    driver: convoy

services:
  redmine:
    ...
    # use persinstent volume for file uploads
    # make public folder accessable for nginx
    volumes:
      - redmine_files:/usr/src/redmine/files
      - redmine_public:/usr/src/redmine/public

  nginx:
    ...
    # access public volume from redmine
    volumes:
      - redmine_public:/usr/src/redmine/public
```

galera

seed.yml:

```
version: '3'

networks:
  backend:
    driver: overlay

services:
  galera-seed:
    image: mariadb-galera-swarm
    build:
      context: galera
      args:
        MYSQL_ROOT_PASSWORD: s3cr3t
    environment:
      - XTRABACKUP_PASSWORD=s3cr3t
      - CLUSTER_NAME=db
      - MYSQL_ROOT_PASSWORD=s3cr3t
      - MYSQL_DATABASE=redmine
      - MYSQL_USER=redmine
      - MYSQL_PASSWORD=r3dmin3
    networks:
      - backend
    deploy:
      replicas: 1
      placement:
        constraints:
          - node.role == manager
          - node.hostname == birthday1
    stop_grace_period: 30s
    command: seed
```

gelara.yml:

```
version: '3'

networks:
  backend:
    driver: overlay

services:
  db:
    image: colinmollenhour/mariadb-galera-swarm
    #build:
    #  context: galera
    #  args:
    #    MYSQL_ROOT_PASSWORD: s3cr3t
    environment:
      - XTRABACKUP_PASSWORD=s3cr3t
      - CLUSTER_NAME=db
      - MYSQL_ROOT_PASSWORD=s3cr3t
      - MYSQL_DATABASE=redmine
      - MYSQL_USER=redmine
      - MYSQL_PASSWORD=r3dmin3
      - GCOMM_MINIMUM=2
    networks:
      - backend
    depends_on:
      - galera-seed
    deploy:
      replicas: 2
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure
    stop_grace_period: 30s
    command: node tasks.galera-seed,tasks.db
```

redmine.yml:

```
version: '3'

networks:
  backend:
    driver: overlay

volumes:
  # persistent volume for redmine uploads
  redmine_files:
    driver: convoy
  # persistent volume for redmine public
  redmine_public:
    driver: convoy

services:
  redmine:
    # use redmine:latest image from docker hub
    image: birthday1:5000/redmine
    # setup redmine environment
    # see https://hub.docker.com/_/redmine/
    environment:
      - REDMINE_DB_MYSQL=bday_db
      - REDMINE_DB_USERNAME=redmine
      - REDMINE_DB_PASSWORD=r3dmin3
    depends_on:
      - db
    # use persinstent volume for file uploads
    # make public folder accessable for nginx
    volumes:
      - redmine_files:/usr/src/redmine/files
      - redmine_public:/usr/src/redmine/public
    networks:
      - backend
    deploy:
      replicas: 1
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost:3000/ || exit 1"]
      interval: 1m
      timeout: 10s
      retries: 3
  nginx:
    # build nginx with Dockerfile in ./nginx
    # see https://hub.docker.com/_/nginx/
    image: birthday1:5000/nginx
    # map localhost:8080 to nginx:80
    ports:
      - 80:80
    depends_on:
      - redmine
    # access public volume from redmine
    volumes:
      - redmine_public:/usr/src/redmine/public
    networks:
      - backend
    depends_on:
      - redmine
    deploy:
      mode: global
      #replicas: 3
      update_config:
        parallelism: 2
        delay: 10s
      restart_policy:
        condition: on-failure
    healthcheck:
      test: ["CMD-SHELL", "curl -f http://localhost/ || exit 1"]
      interval: 1m
      timeout: 10s
      retries: 3
```

Deployment:

```
docker -f seed.yml build galera-seed
docker stack deploy -c seed.yml bday
```

```
watch docker stack ps bday
```

```
docker exec bday_galera-seed... /bin/bash
> mysql redmine < /docker-entrypoint-initdb.d/redmine.dmp.sql
```

```
docker stack deploy -c galera.yml bday
```

```
watch docker stack ps bday
```

```
docker service rm bday_galera-seed
docker service scale bday_db=3
```

```
watch docker stack ps bday
```

```
docker stack deploy -c redmine.yml bday
```

```
watch docker stack ps bday
```

portainer

weave scope