Postgres Using Swarm

- Install Docker Swarm
- Install Postgres

Install Docker Swarm

تنظیمات فایروال را برای همه دون ها ست میکنیم:

```
firewall-cmd --add-port=2376/tcp --permanent
firewall-cmd --add-port=2377/tcp --permanent
firewall-cmd --add-port=7946/tcp --permanent
firewall-cmd --add-port=7946/udp --permanent
firewall-cmd --add-port=4789/udp --permanent
firewall-cmd --reload
```

برای همه دون ها HostName مناسب ست میکنیم

```
hostnamectl set-hostname {NEW_NAME}
```

یک دون را مستر در نظر میگیریم و در دون مستر دستور زیر را میزنیم:

```
docker swarm init
```

دستور زیر را در دون Manager فعلی اجرا می کنیم تا یک کامند دارای توکن تولید کند و مطابق مراحلی که نتیجه کامند میدهد پیش می رویم.

```
docker swarm join-token manager
```

در این مرحله برچسب های دون‌ها در کلاستر Swarm را براساس HOSTNAME آن ها، تنظیم می کنیم. برچسب گذاری کمک می کند برای استقرار سرویس ها در دون صحیح، براساس برچسب هر دون تصمیم گیری کنیم؛ در واقع برچسب هایی که در این مرحله برای دون‌های عضو کلاستر تعیین می شود، در زمان بندی و استقرار WORKLOAD ها (برای مثال Postgres) روی دون‌ها مورد استفاده قرار می گیرد.

```
docker node update --label-add region={LABEL} {HOSTNAME}
```

جهت مشاهده لیست دون‌ها به همراه label ها از دستور زیر استفاده می شود.

```
docker node ls -q | xargs docker node inspect -f '{{ .ID }} [{{ .Description.Hostname }}]: {{
.Spec.Labels }} - {{ .Status.Addr }}'
```

Install Postgres

دایرکتوری زیر را در همه نود های دیتابیس ایجاد میکنیم:

```
mkdir -p /home/postgres/data
```

یک فایل docker-compose.yaml ایجاد کرده بصورت زیر: (درصورتی که تعداد سرور ها بیش از 2 عدد است فایل را تغییر دهید)

```
version: "3.6"
services:
  zk1:
    image: {REGISTRY_IP}:{REGISTRY_PORT}/bitnami/zookeeper:3.9.1
    deploy:
      placement:
        constraints:
          - "node.labels.region==db-1"
    ports:
      - target: 2181
        published: 2181
        mode: host
      - target: 2888
        published: 2888
        mode: host
      - target: 3888
        published: 3888
        mode: host
    networks:
      db_net: null
    environment:
      - ALLOW_ANONYMOUS_LOGIN=yes
      - ZOO_SERVER_ID=1
      - ZOO_SERVERS=0.0.0.0:2888:3888,{SECOND_SERVER_IP}:2888:3888
  zk2:
    image: {REGISTRY_IP}:{REGISTRY_PORT}/bitnami/zookeeper:3.9.1
    deploy:
      placement:
        constraints:
```

```
        - "node.labels.region==db-2"
    ports:
      - target: 2181
        published: 2181
        mode: host
      - target: 2888
        published: 2888
        mode: host
      - target: 3888
        published: 3888
        mode: host
    networks:
      db_net: null
    environment:
      - ALLOW_ANONYMOUS_LOGIN=yes
      - ZOO_SERVER_ID=2
      - ZOO_SERVERS={FIRST_SERVER_IP}:2888:3888,0.0.0.0:2888:3888

  pgnode1:
    image: {REGISTRY_IP}:{REGISTRY_PORT}/ghcr.io/zalando/spilo-15:3.0-p1
    ports:
      - target: 5432
        published: 5432
        mode: host
    deploy:
      replicas: 1
      placement:
        constraints:
          - "node.labels.region==db-1"
    environment:
      ZOOKEEPER_HOSTS: {FISRT_SERVER_IP}:2181,{SECOND_SERVER_IP}:2181
      PGPASSWORD_STANDBY: {PASSWORD}
      PGPASSWORD_ADMIN: {PASSWORD}
      PGPASSWORD_SUPERUSER: {PASSWORD}
      SCOPE: pgCluster
    networks:
      db_net: null
    volumes:
      - /home/postgres/data:/home/postgres/pgdata
```

```
pgnode2:

    image: {REGISTRY_IP}:{REGISTRY_PORT}/ghcr.io/zalando/spilo-15:3.0-p1
    ports:
      - target: 5432
        published: 5432
        mode: host
    deploy:
      replicas: 1
      placement:
        constraints:
          - "node.labels.region==db-2"
    environment:
      ZOOKEEPER_HOSTS: {FISRT_SERVER_IP}:2181,{SECOND_SERVER_IP}:2181
      PGPASSWORD_STANDBY: {PASSWORD}
      PGPASSWORD_ADMIN: {PASSWORD}
      PGPASSWORD_SUPERUSER: {PASSWORD}
      SCOPE: pgCluster
    networks:
      db_net: null
    volumes:
      - /home/postgres/data:/home/postgres/pgdata

networks:
  db_net:
    name: mavara-pgsql-network
    external: true
    driver: overlay
```

و {SECOND_SERVER_IP} ، {FIRST_SERVER_IP} ، {REGISTRY_PORT} ، {REGISTRY_IP} ری‌داق‌م دی‌ن‌ک م‌ی‌ظ‌ن‌ت ح‌ی‌ح‌ص ری‌داق‌م س‌اس‌ا رب ار {PASSWORD}

م‌ی‌ن‌ک‌ی‌م ارج‌ا ار ری‌ز ت‌اروت‌س‌د Manager Swarm دون ی‌ور

```
unzip postgres-cluster.zip
cd postgres-cluster
chmod +x install-postgres-cluster.sh
./install-postgres-cluster.sh {REGISTRY_IP}:{REGISTRY_PORT}
```

دوج‌وم docker-compose.yaml ل‌ی‌اف ه‌ک ی‌روت‌ک‌ری‌اد رد ،ال‌اب روت‌س‌د ن‌دوب زی‌م‌آ ت‌ی‌ق‌ف‌وم ت‌روص رد م‌ی‌ن‌ک‌ی‌م ارج‌ا ار ری‌ز ت‌اروت‌س‌د ت‌س‌ا

```
docker network create -d overlay --attachable mavara-pgsql-network
docker stack deploy --compose-file docker-compose.yml the-pgsql-stack
```

کامند های زیر به بررسی وضعیت کلاستر PostgreSQL می پردازد.

```
docker service ls
docker stack services the-pgsql-stack
```

دستورات زیر برای بررسی لاگ سرویس ها میباشد (شماره instance ها را در آخر نام سرویس تغییر دهید)

```
docker service logs the-pgsql-stack_zk1
docker service logs the-pgsql-stack_pgnode1
```

جهت پایین آوردن کلاستر، کامند `docker stack rm` از طریق یکی دون Manager اجرا می شود

```
docker stack rm the-pgsql-stack
```