

title: Docker 启动 Redis

date: 2018-11-09

tags: Docker, Database

**description: 使用 Docker 在 Node 服务中添加 Redis。**

## 构建需要的镜像

```
docker pull redis
docker pull node
# 启动 node
docker run -it node /bin/bash
npm install pm2 -g
npm install cnpm -g --registry=https://registry.npm.taobao.org
exit

# 发布镜像
docker login
# 把 Container 提交为 Images
docker commit uuid wkl/node_pm2
# 查看 Images 列表
docker images node_pm2
# 把镜像提交到云上
docker push wkl/node_pm2
```

## 构建执行环境

1. 通过 Redis 镜像启动一个 Redis 的 Container

```
docker run --name redis-server -d redis redis-server --appendonly yes
```

2. 创建 app.js, 启动 8000 端口监听

```
var express = require('express');
var redis = require('redis');

var app = express();

// 从环境变量中读取 Redis 服务器的 IP 地址
var redisHost = process.env['REDIS_PORT_6379_TCP_ADDR'];
var redisPort = process.env['REDIS_PORT_6379_TCP_PORT'];

var redisClient = redis.createClient(redisPort, redisHost);

app.get('/', function(req, res) {
  console.log('get request');
  redisClient.get('access_count', function(err, countNum) {
    if (err) {
      return res.send('get access count error');
    }
    if (!countNum) {
      countNum = 1;
    } else {
      countNum = parseInt(countNum) + 1;
    }
    redisClient.set('access_count', countNum, function(err) {
      if (err) {
        return res.send('set access count error');
      }
      res.send(countNum.toString());
    });
  });
});
```

```
app.listen(8000);
```

### 3. 启动一个 Container 安装依赖包

```
docker run --rm \  
-itv /current_dir/express_test:/var/node/docker_node \  
-w /var/node/docker_node \  
wkl/node_pm2 \  
npm install
```

-w 参数表示命令执行的当前工作目录，屏幕会打印依赖包的安装过程。等依赖安装完成，这个 Container 会自动退出。如果出现 EACCESS 的权限错误，可以执行命令许可 SELinux 的工作状态。这个只是临时修改，重启系统会恢复

```
su -c "setenforce 0"
```

### 4. 启动一个 Container 运行程序，要求这个 Container 有以下要求：

- 端口映射
- 文件挂载log、code
- 加载 Redis 的 Container

```
docker run -d \  
--name 'nodeCount' \  
-p 8000:8000 \  
-v /current_dir/express_test:/var/node/docker_node \  
-v /current_dir/log/pm2:/root/.pm2/logs \  
--link redis-server:redis \  
-w /var/node/docker_node/ \  
wkl/node_pm2 \  
pm2 start app.js
```

执行 `docker ps` 发现这个 Container 并没有启动，这是利用 pm2 的守护进程启动了应用，Container 认为进程已经运行结束。可以做一些调整

```
docker run -d \  
--name 'nodeCount' \  
-p 8000:8000 \  
-v /current_dir/express_test:/var/node/docker_node \  
-v /current_dir/log/pm2:/root/.pm2/logs \  
--link redis-server:redis \  
-w /var/node/docker_node/ \  
wkl/node_pm2 \  
pm2 start --no-daemon app.js
```

在执行 `docker ps` ,可以看到对应的 Container

### 5. 使用 Nginx 做反向代理 待续。。。