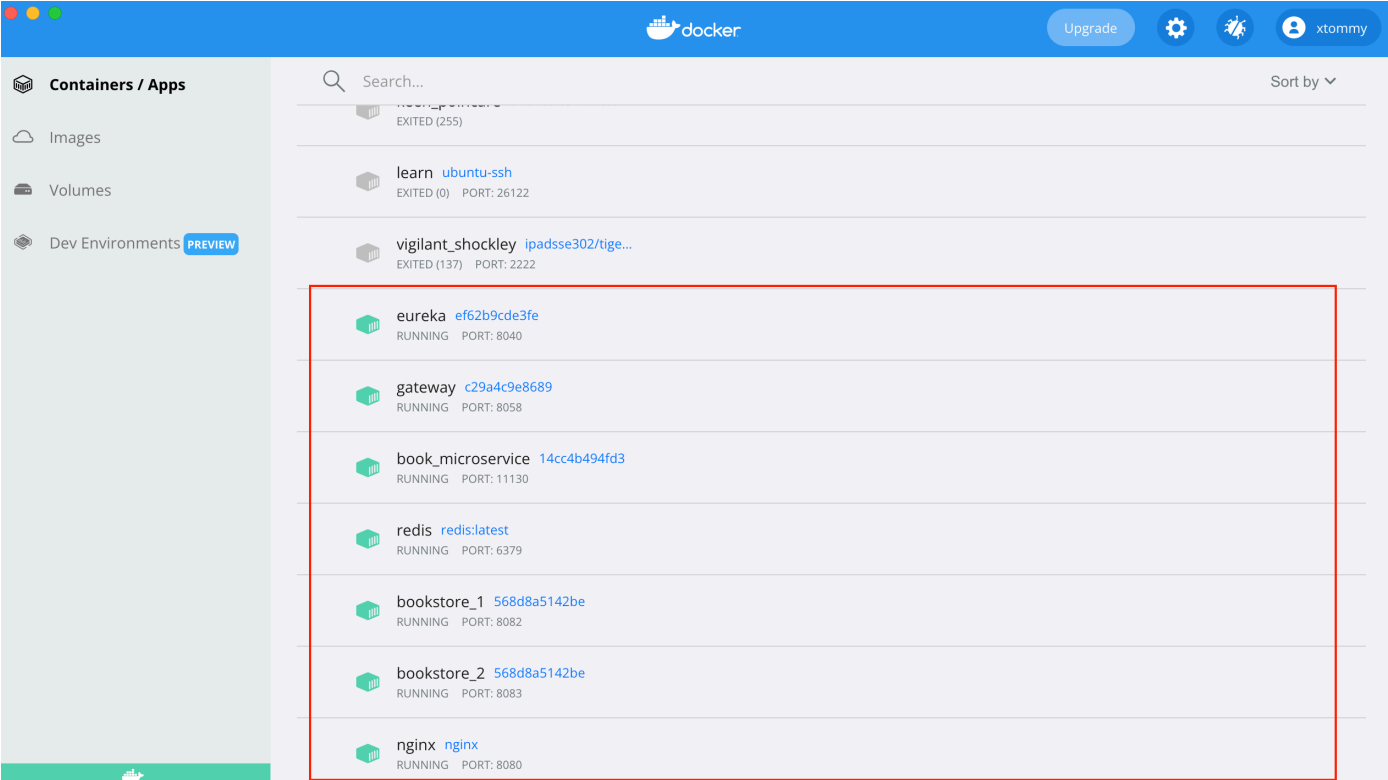


# 作业十：Clustering & Docker

519021910913 黄喆敏

根据作业要求，我们选用nginx进行负载均衡，包括1个nginx实例，1个Redis实例，2个Tomcat后端实例，以及对应的微服务实例，并将后端实例注册到Eureka注册中心上。

实例名称及映射的端口号如下图所示：



## 配置方法

### 1. bookstore实例

我们采用Dockerfile的方法，构建镜像。Dockerfile内容如下所示：

```
FROM tomcat:latest

#暴露8082端口
EXPOSE 8082

#声明挂载点
VOLUME /tmp

#系统编码
ENV LANG=C.UTF-8 LC_ALL=C.UTF-8

#将maven打包的jar文件复制到镜像中，同时名称改为app.jar
ADD bookstore-0.0.1-SNAPSHOT.jar app.jar
```

#创建新文件，同时修改文件属性

```
RUN sh -c 'touch /app.jar'
```

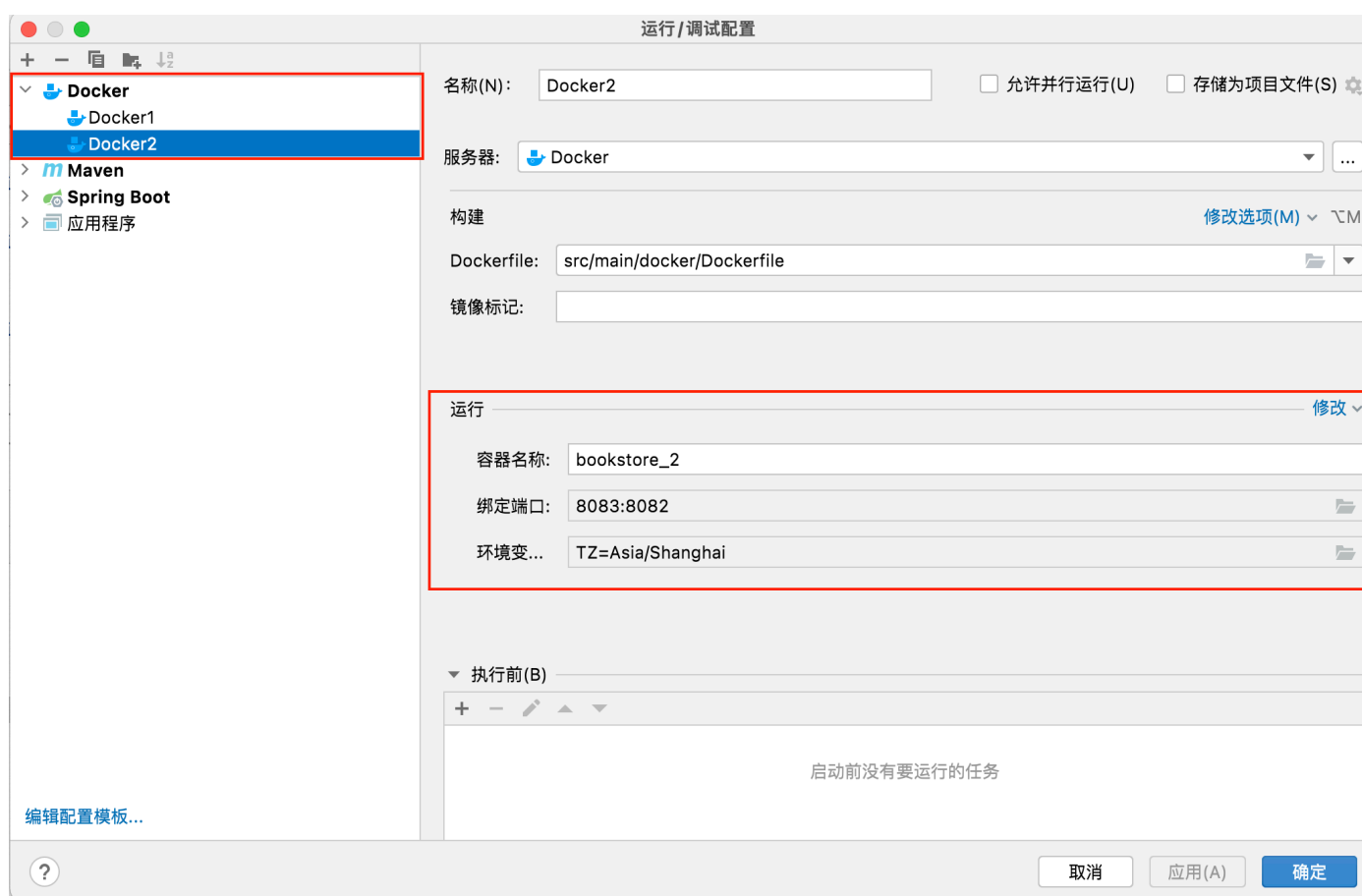
#启动容器后的运行指令

```
ENTRYPOINT ["java", "-jar", "/app.jar"]
```

由于数据库没有部署在容器内，因此我们将application.properties的数据库路径由localhost改为**host.docker.internal**，便于容器访问宿主机端口。

接下来用maven进行打包。需要注意的是，websocket在maven打包时会与spring boot内嵌的tomcat产生冲突，无法通过单元测试。因此我们采用 `mvn clean package -DskipTests` 的命令，跳过测试进行编译。

接着，我们在IDEA中启动两个实例，配置如下图所示：



Eureka实例以及微服务实例配置方法与上面相同，因此不再赘述。

## 2. Redis实例

我们采用以下命令，创建Redis实例：

```
docker run -d --name redis -p 6379:6379 redis:latest redis-server --appendonly yes
```

Redis在宿主机以及容器中均为6379端口，且开启了数据持久化。

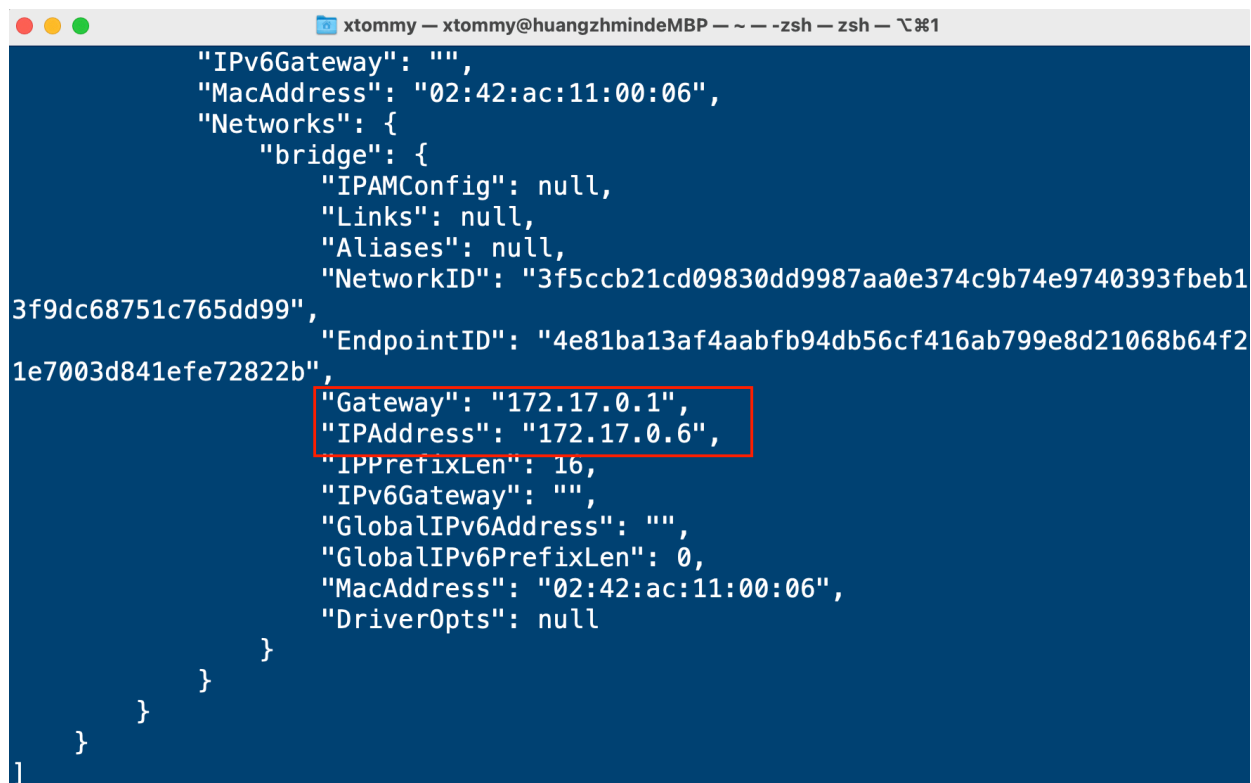
在bookstore实例中，我们主要通过 `com.bookstore.config.RedisSessionConfig` 实现Session的共享。Spring会将session的相关信息储存到Redis实例中，解决了负载均衡所带来的粘滞会话问题。

### 3. Nginx实例

首先我们采用以下命令，创建一个测试用的Nginx实例，获得默认的 `nginx.conf` 配置文件。

```
docker run --name test -d nginx
docker cp test:/etc/nginx/nginx.conf /Users/xtommy/docker/nginx/conf
```

通过 `docker inspect` 命令，我们可以获得bookstore\_1和bookstore\_2的IP地址，分别为172.17.0.6、172.17.0.7。然后，我们根据ip地址，修改配置文件的代理规则。



```
xtommy — xtommy@huangzhmindeMBP — — -zsh — zsh — ￼%1
  "IPv6Gateway": "",
  "MacAddress": "02:42:ac:11:00:06",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "NetworkID": "3f5ccb21cd09830dd9987aa0e374c9b74e9740393fbeb1
3f9dc68751c765dd99",
      "EndpointID": "4e81ba13af4aabfb94db56cf416ab799e8d21068b64f2
1e7003d841efe72822b",
      "Gateway": "172.17.0.1",
      "IPAddress": "172.17.0.6",
      "IPPrefixLen": 16,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "MacAddress": "02:42:ac:11:00:06",
      "DriverOpts": null
    }
  }
}
```

配置文件中的主要修改如下所示，设置权重为1:1。

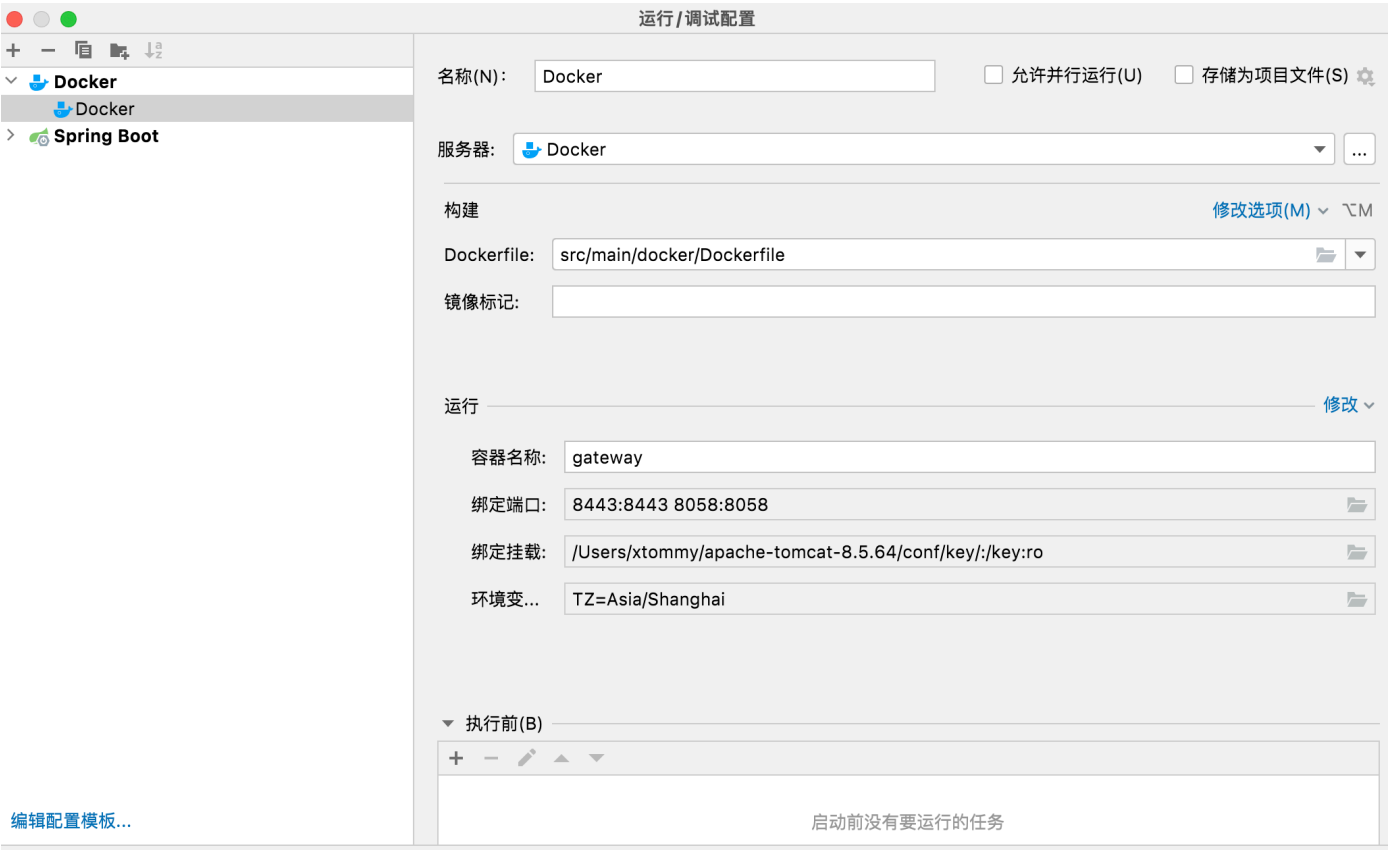
```
upstream pancm{
    #ip_hash;
    #least_conn;
    server 172.17.0.6:8082; # bookstore_1
    server 172.17.0.7:8082; # bookstore_2
}
server {
    listen 80;
    server_name localhost;
    location / {
        root html;
        proxy_pass http://pancm;
        index index.html index.htm;
    }
}
```

最后，采用以下命令，创建Nginx实例。我们将修改后的本地配置文件挂载到docker中。同时使用--link链接Nginx和两个bookstore实例，使得它们之间可以互相通信。

```
docker run -it --name nginx -v /Users/xtommy/docker/nginx/conf/nginx.conf:/etc/nginx/nginx.conf:ro -d -p 8080:80 --link bookstore_1:bookstore_1 --link bookstore_2:bookstore_2 nginx
```

## 4. gateway实例

由于之前配置了https访问，因此在启动gateway时，我们将http和https所对应的两个端口分别进行映射；同时将本机的证书所在目录挂载到容器中，并设为只读。配置如下图所示：



## 配置结果

- 访问localhost:8080，可以看到Nginx的默认页面，说明Nginx配置成功。

# Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

- 接着采用Postman，向8080端口多次模拟登录请求，可以看到请求成功；且普通的请求也可成功响应。

New Collection / 登录请求

POST http://localhost:8080/login

Body

```
1 {
2   "username": "admin",
3   "userPassword": "admin"
4 }
```

Status: 200 OK Time: 440 ms Size: 418 B

Body

```
1 {
2   "status": 1,
3   "message": "登录成功!",
4   "data": {
5     "userId": 2,
6     "userType": 1,
7     "username": "admin"
8   }
9 }
```

New Collection / New Request

GET http://localhost:8080/getBooks

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 1304 ms Size: 9.13 KB

Body

```
1 {
2   "bookId": 1,
3   "name": "新教伦理与资本主义精神",
4   "author": "[德] 马克斯·韦伯",
5   "price": 15.00,
6   "isbn": "9787506393102",
7   "inventory": 1005,
8   "description": "《新教伦理与资本主义精神》是马克斯·韦伯最著名的著作之一。在这部作品中，韦伯提出了一个知名的论点：新教教徒的思想影响了资本主义的发展。宗教教徒往往排斥世俗的事务，尤其是经济成就上的追求，但为什么新教教徒却是例外？韦伯在该书中论述宗教观念（新教伦理）与隐藏在资本主义发展背后的某种心理驱动力（资本主义精神）之间的关系。",
9   "image": "https://cdn.jsdelivr.net/gh/xtommy-1/Images@master/book_1.4zmxniwd0nsw.jpg",
10  "type": "哲学/宗教类",
11  "brief": "组织理论之父 现代社会学经典",
12  "enabled": true
13 }
14 }
```

- 查看两个实例的日志，可以看到两个实例分别接收了一次登录请求。由于登录的是一个账号，因此他们共享了同一个session。

The first screenshot shows the logs for the `/bookstore_1` Dockerfile. It displays a successful login for user 'admin' with ID 2, resulting in a session ID: `a40a6f86-db0f-4679-801b-5b0fd1d26149`. The session attributes are `userId: 2`, `username: admin`, and `userType: 1`.

The second screenshot shows the logs for the `/bookstore_2` Dockerfile. It displays a successful login for user 'admin' with ID 2, resulting in a session ID: `a40a6f86-db0f-4679-801b-5b0fd1d26149`. The session attributes are `userId: 2`, `username: admin`, and `userType: 1`. This confirms that both instances shared the same session.

- 可以看到，session相关的信息均存储到了Redis中。由于之前访问了书籍相关信息，因此bookList也被缓存到了Redis中。

The screenshot shows the Redis Desktop Manager interface. On the left, the database structure is visible, including `db0 (4)` with keys like `bookList`, `spring (3)`, `session (3)`, `expirations (1)`, and `sessions (2)`. The `sessions` key is highlighted, showing a list of session IDs.

On the right, the details for the selected session are shown. The `HASH` table contains the following data:

row	key	value
1	maxInactiveInterval	2147483647
2	sessionAttr:username	admin
3	sessionAttr:userId	2
4	lastAccessedTime	1638286200000
5	sessionAttr:userType	1
6	creationTime	1638286200000

- 最后访问localhost:8040。可以看到，gateway、两个bookstore实例、微服务实例均成功注册。

## System Status

Environment	test	Current time	2021-11-30T22:29:57 +0800
Data center	default	Uptime	02:20
		Lease expiration enabled	true
		Renews threshold	8
		Renews (last min)	16

## DS Replicas

localhost
-----------

## Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
BOOKSERVICE	n/a (1)	(1)	UP (1) - d4a392cc0c33:bookservice:11130
BOOKSTORE	n/a (2)	(2)	UP (2) - 4a1752177530:bookstore:8082 , 4aab4afc9608:bookstore:8082
GATEWAY	n/a (1)	(1)	UP (1) - 22435b75d120:gateway:8443

## References

- [1] <https://blog.csdn.net/u010433704/article/details/99945557>
- [2] <https://www.cnblogs.com/dotnet261010/p/12596185.html>
- [3] [https://blog.csdn.net/qq\\_42114918/article/details/85238011](https://blog.csdn.net/qq_42114918/article/details/85238011)