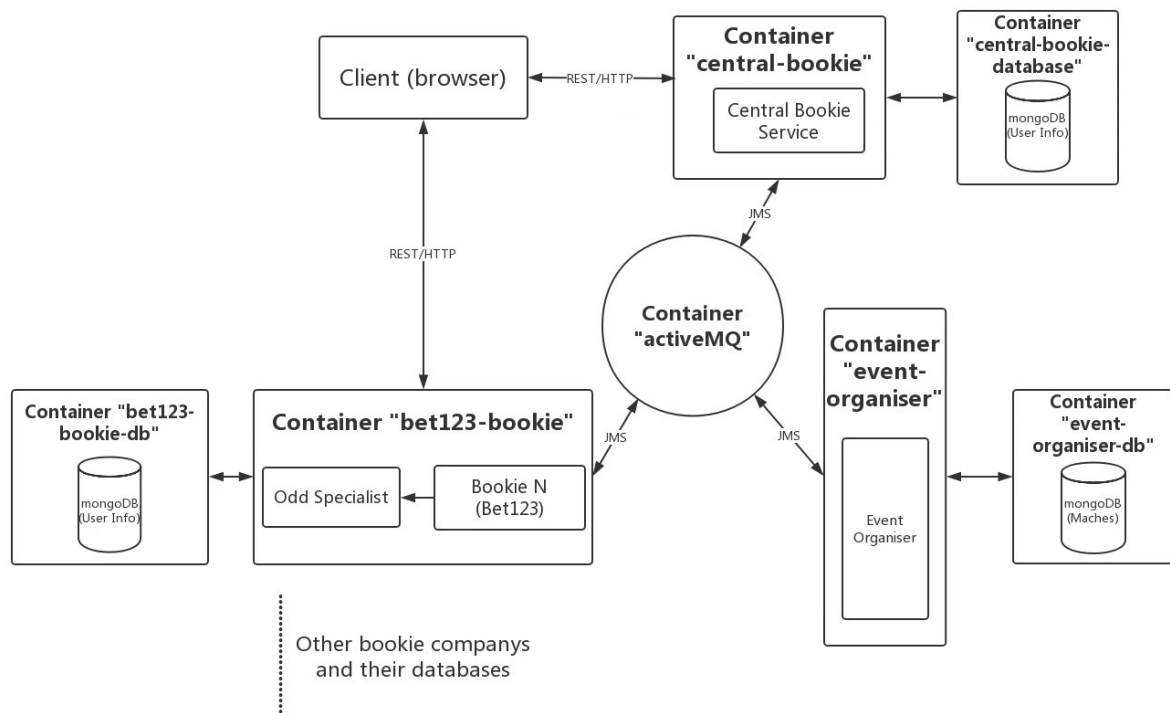


Dockerize the Project

In order to demonstrate that different services in our project can be hosted on different machines and therefore be horizontally scalable, we used docker and docker network to deploy our system.

Architecture

There are seven containers we need to open in the current scenario. Which should be three different services: "central-bookie", "bet123-bookie" and "event-organiser". Each one of these has a separate mongo database connected. These databases are also wrapped in their own container. Despite that, we also need to create a container with a activeMQ inside. The whole architecture is shown down below.



Notice that all the containers will be run on the same network with different names and IPs. Furthermore, it demonstrates that our system could be deployed and run through a network, which is distributed.

Deployment

In order to deploy this system on your computer, you first need to make sure that you install docker or docker hub successfully. In the meantime, it is better that you can monitor the containers' log (by using toolbox or IDEA plugin) since our scripts let all containers run in the background without showing any information inside these containers.

Once you installed docker on your computer, you need to get into the folder "docker-network-cluster" and you will see a few folders and files including three separate projects, two shell scripts, and a folder including some matches information.

Bet123	EventOrganiser	db	demo-stop.sh
CentralBookie	demo-start.sh		

The database content is used to published some matches info into the event organiser's database thus users can find some matches to bet on instead of nothing.

To run the docker cluster, simply run the "**demo-start.sh**".

```
./demo-start.sh
```

It will create a docker network names "bookie-system" and run seven docker containers automatically.

You will need to wait a few moments until all three applications successfully started.

Type "**localhost:8081**" on a browser to view our homepage!

To stop and delete these containers, simply use:

```
./demo-stop.sh
```

Notice

1. The docker system is an older version of the main project due to some developing time issues
2. Inside the "demo-start.sh" script we use "**work_path=\$(pwd)**" to get your current absolute directory, the "db" folder under it and set up the event-organiser's database, which might be not correct in your computer. If the deployment fails because of this, try to modify it so that the script could find the "db" folder.
3. You can view our docker repository [here](#).