Retail Supply Chain Eco-System Build & Run Guide

The system is comprised of two parts: the front-end architecture and the back end. The following commands are tested on **ubuntu linux 16.04LTS**.

The Front-end

The front-end is compiled with **yarn**. Due to the huge size of the project, you must set two additional nodejs parameters.

* NODE\_OPTIONS=--max-old-space-size=10230，increase the compiled content, or install and download increase-memory-limit
* PUPPETEER\_SKIP\_CHROMIUM\_DOWNLOAD=1，don't download chrome to avoid taking too long to download

Then execute command:

cd retailscm-biz-suite/bizui/ && yarn install && yarn build

The download time varies depending on the network, and the compilation time will take about 300 seconds to 700 seconds. This step requires a high performance CPU.

In the dist directory under the bizui directory, there will be all the js files and other files that need to be deployed. It can be deployed anywhere. Using CDNs can greatly help the response speed.

Backend

Backend has reverse proxy server ngnix, servlet container Resin or Tomcat (late for Spring Boot), database server MySQL, cache server Redis, message server kafka, multi-level rights management needs graph database arrangodb, external email server, Ali cloud SMS server, OSS server, Aurora app message push server, blockchain HyperLedger fabric node.

Download **Resin**

<https://caucho.com/products/resin/download/3-1/gpl>

Install docker and accelerate it by domestic mirror image(Please ignore when you out of China)

sudo curl -sSL https://get.daocloud.io/docker | sh

http://84763bc6.m.daocloud.io

sudo groupadd docker

sudo usermod -aG docker $USER

Install **MYSQL** and **Redis**

The system needs redis and mysql to run, both are installed by docker, the command is as follows:

docker run -e MYSQL\_ROOT\_PASSWORD=0254891276 -p 3306:3306 --name demo\_db mysql:5.7

docker run -d --name demo\_redis -p 6379:6379 redis

Please note that the default character set of mysql5.7 is not utf8mb4, needs to modify the configuration to support utf8mb4.

In-container file /etc/mysql/my.cnf

[client]

default-character-set = utf8mb4

[mysql]

default-character-set = utf8mb4

[mysqld]

character-set-client-handshake = FALSE

character-set-server = utf8mb4

collation-server = utf8mb4\_unicode\_ci

time\_zone =+08:00

**Redis** is very simple, just run it.

Compilation

The java project uses gradle to compile. For fast development, we just compile the java file into class, and the other directory structure remains unchanged. It is recommended to set the output directory directly to classes and use the development mode of resin, so that when the class changes, Resin will automatically reload the new class without recompiling and starting. The development experience is similar to writing PHP.

Use the latest gradle 5.1+, sdk install gradle 5.1

cd retailscm-biz-suite/bizcore&& gradle classes

It will take about 10 seconds to 20 seconds to get the compiled classes, which is in the directory WEB-INF/classes

Configuring **nginx**

This step is very simple, copy this file to /etc/nginx/sites-enabled/demo, then service ngnix start

server {

gzip on;

gzip\_disable "msie6";

gzip\_comp\_level 6;

gzip\_min\_length 1100;

gzip\_buffers 16 8k;

gzip\_proxied any;

gzip\_types

text/plain

text/css

text/js

text/xml

text/javascript

application/javascript

application/x-javascript

application/json

application/xml

application/rss+xml

image/svg+xml;

# SSL configuration

#

# listen 443 ssl default\_server;

# listen [::]:443 ssl default\_server;

#

# Note: You should disable gzip for SSL traffic.

# See: https://bugs.debian.org/773332

#

# Read up on ssl\_ciphers to ensure a secure configuration.

# See: https://bugs.debian.org/765782

#

# Self signed certs generated by the ssl-cert package

# Don't use them in a production server!

#

# include snippets/snakeoil.conf;

root /var/www/html;

# Add index.php to the list if you are using PHP

index index.html index.htm index.nginx-debian.html;

server\_name demo.doublechaintech.com;

location / {

proxy\_pass http://127.0.0.1:8080;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Host $host;

proxy\_set\_header X-Forwarded-Server $host;

proxy\_set\_header X-Forwarded-Port 80;

proxy\_set\_header X-Forwarded-Proto http;

}

listen 443 ssl; # managed by Certbot

ssl\_certificate /etc/letsencrypt/live/demo.doublechaintech.com/fullchain.pem; # managed by Certbot

ssl\_certificate\_key /etc/letsencrypt/live/demo.doublechaintech.com/privkey.pem; # managed by Certbot

include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot

ssl\_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}

**Tomcat** Container(Not Supported)

The bizcore/WEB-INF/web.xml path matching rule needs to be modified, and nothing else needs to be changed

SpringBoot is supported