DEVOPS EASY LEARNING

S5GROUP5
BrainCells
(E-commerce)

APPLICATION PRESENTATION

The E-commerce

Name: Explorer

Deployment strategie: **Docker-compose**

DEVOPS EASY LEARNING

GOAL:

- Deploy The Explorer application using docker-compose

INFO:

- Customer Name: **Dragon-ball Z**

Services for Explorer Application

Frond End (02):

- Wordpress
- Frontend(server)

Middle Services OR API (02):

- Worker
- Storefront

Databases 06:

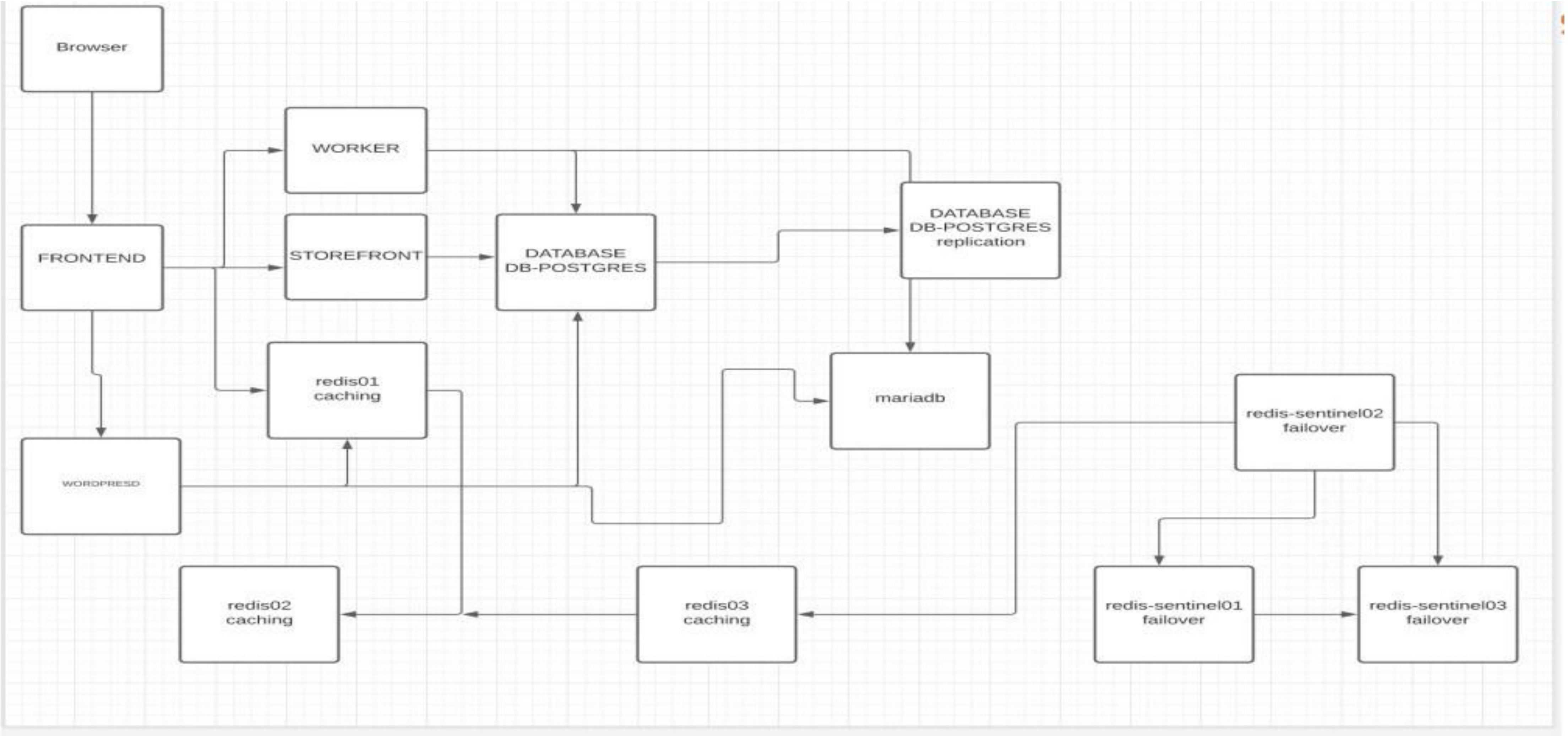
- Redis catching (03) ==>> For session catching
- Postgres (02)
- Mariadb (01)

Redis Failover Management:

- Sentinel (03)

DEVOPS EASY LEARNING

FLOW



DEVOPS EASY LEARNING

Requirement from customer

High availability
bi-weekly release
shopping ready e-commerce platform

Requirement from Development team

entire application code
all necessary instructions required to get the application working

Requirement from DevOps team

1. deploy the current application
2. B- ensure high availability

Instructions from Development team

- The code is available on company s3bucket

- Perform the following command to access it
  - Wget https://group5-braincells.s3.amazonaws.com/E-commerce.zip

- Then cd inside ecommerce directory
- All work are to be done inside this is directory

Instructions from Development team

The who application is made of the following services:

- Server:

expose port 3000

suggested map port 3000

The following command should be run once the container start ["./wait-for-it.sh", "database:5432", "--", "yarn", "run:server"]

volume should be map to /usr/src/app

Environment variable: DATABASE_HOST=database ; DATABASE_PORT=5432 (preferable called inside the compose file but we leave this to your discretion)

This service should not start before the database and worker service

The base image is: **devopseasylearning2021/s5dylan:frontends**

Instructions from Development team

The who application is made of the following services:

- **Worker**

The following command should be run once the container start ["./wait-for-it.sh", "database:5432", "--", "yarn", "run:worker"]

volume should be map to /usr/src/app

Environment variable: DATABASE_HOST=database ; DATABASE_PORT=5432 (preferable called inside the compose file but we leave this to your discretion)

This service should not start before the database service

The base image is: **devopseasylearning2021/s5dylan:frontends**

- Storefront

expose port 4000

suggested map port 4000

Environment variable: SERVER_API_HOST=http://server, SERVER_API_PORT=3000, SERVER_API_PATH=shop-api (preferable called inside the compose file but we leave this to your discretion).

The base image is : **devopseasylearning2021/s5dylan:storefront**

Instructions from Development team

The who application is made of the following services:

- **Database:**

recommended map port 5432
environment variables: POSTGRES_PASSWORD=password ; POSTGRES_DB=vendure
Image **postgres**
Do the same configuration for the replications the numbers of replication is up to you.

- mariadb:

Environment variable: MYSQL_ROOT_PASSWORD=password (preferable called inside the compose file but we leave this to your discretion)
base image **mariadb**
please map to port 3306

Instructions from Development team

The who application is made of the following services:

- **wordpress:**

should not start before mariadb service

for best performance use the following connection mariadb:mysql

map to port 80 and 443

Environment variable:

MONEI_OPERATION_MODE=live
MONEI_TEST_CHANNEL_ID=test_channel_id
MONEI_TEST_USER_ID=test_user_id
MONEI_TEST_PASSWORD=test_password
MONEI_LIVE_CHANNEL_ID=live_channel_id
MONEI_LIVE_USER_ID=live_user_id
MONEI_LIVE_PASSWORD=live_password
WORDPRESS_DB_HOST=mariadb:3306
WORDPRESS_DB_PASSWORD=password
WORDPRESS_SITE_TITLE=some_site_title
WORDPRESS_ADMIN_PASSWORD=1234

The base image is : **wordpress:4.7-php7.1-apache**

Create files and mount under /etc/redis/redis.conf inside the container

- file name Redis-0:

```
protected-mode no
port 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

- file name Redis-1:

```
protected-mode no
port 6379
slaveof redis-0 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

- file name Redis-2:

```
protected-mode no
port 6379
slaveof redis-0 6379

#authentication
masterauth a-very-complex-password-here
requirepass a-very-complex-password-here
```

Create files and mount under /etc/redis/sentinel.conf inside the container

Create files : **sentinel-0, sentinel-2, sentinel-3 ,** With The following same contain

```
port 5000
sentinel monitor mymaster redis-0 6379 2
sentinel down-after-milliseconds mymaster 5000
sentinel failover-timeout mymaster 60000
sentinel parallel-syncs mymaster 1
sentinel auth-pass mymaster a-very-complex-password-here
```

Instructions from Development team

For session caching use redis cluster with the following:
- 03 redis nodes with data replication
- 03 redis-sentinels with failover detection

Propose base image :
redis:6.0-alpine

Special instructions

Propose volumes with driver set to local:

- Mariadb_data
- Wordpress_data
- Apache_data
- apache_data

Start redis replication with following command

```
redis-server /etc/redis/redis.conf
```

Start sentinel with following command

```
redis-sentinel /etc/redis/sentinel.conf
```

GOOD JOB

DEVOPS EASY LEARNING