DEVOPS EASY LEARNING

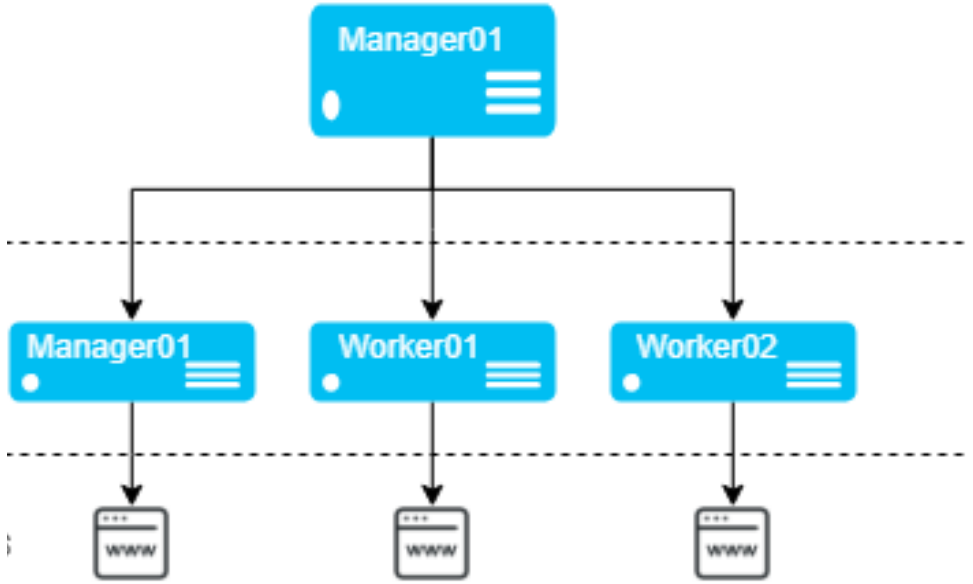
Class review S5GROUP5 BrainCells (DOCKERCOMPOSE/AWS/yaml /microservice)

Questions

1. What is docker compose?
2. What is the difference between docker-compose and docker compose?
3. Can you please give a brief explanation of how a docker compose yaml file is structure?
4. Can you please briefly explain the following service options
    - Command
    - Restart
    - Volume
    - Depend_on
    - Network

Questions

5. Look at the image below, using a scenario approach explain with your word what do you understand by
   - Clustering
   - High Availability
   - Replication



DEVOPS EASY LEARNING

The restaurant app

Name: Odilia

Deployment strategie: **Docker-compose**

GOAL:

- Deploy The Odilia application using docker-compose

INFO:

- Customer Name: McDonald's

INFO:

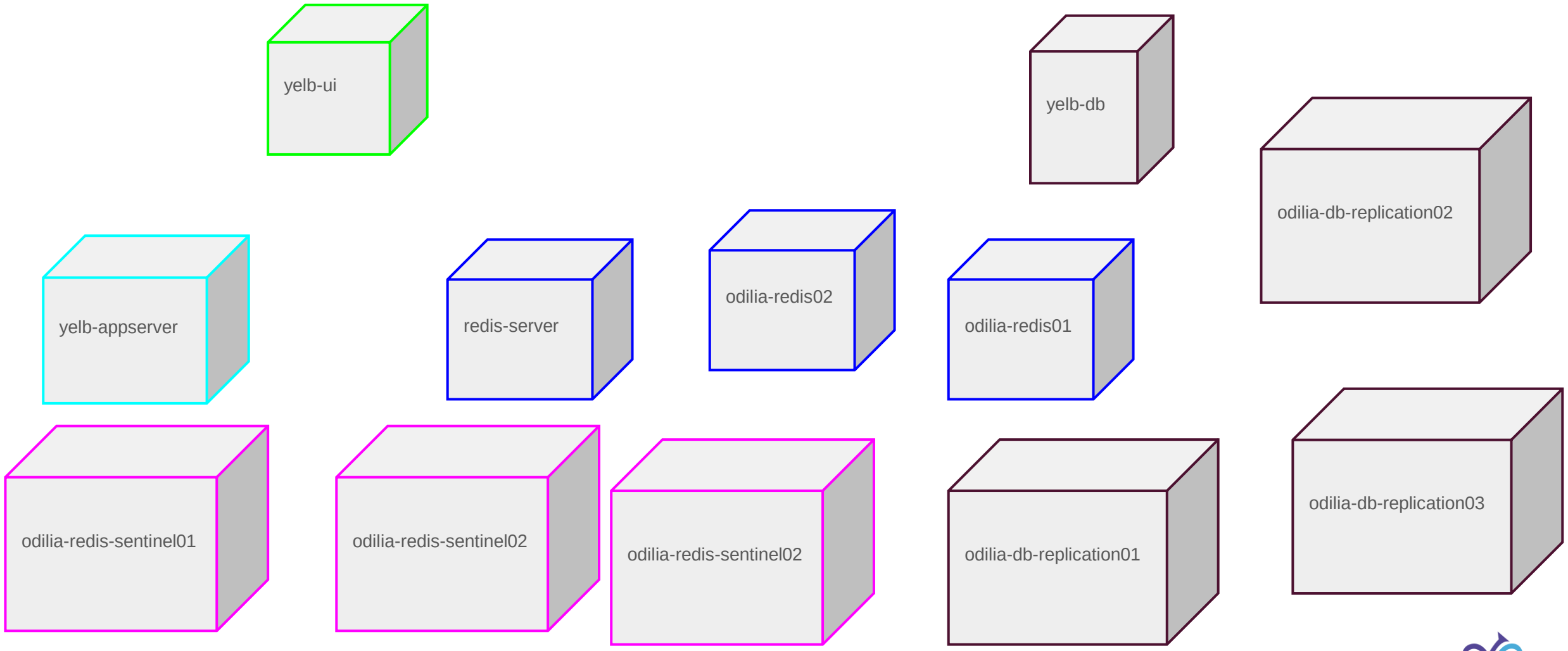
- In an effort to best serve customer and to increase it number of customer McDonald has contracted with EK TECH SOFTWARE SOLUTION to build and application which would allow customers to vote the type of restaurant they visit most of the times
- Odilia allows users to vote on a set of alternatives (restaurants) and dynamically updates pie charts based on number of votes received. In addition to that Odilia keeps track of number of page views as well as it prints the hostname of the yelb-appserver instance serving the API request upon a vote or a page refresh. This allows an individual to demo the application solo, or involving people (e.g. an audience during a presentation) asking them to interact by pointing their browser to the application (which will increase the page count) and voting their favorite restaurant.

INFO:

The Application is made of 12 microservices as follow:

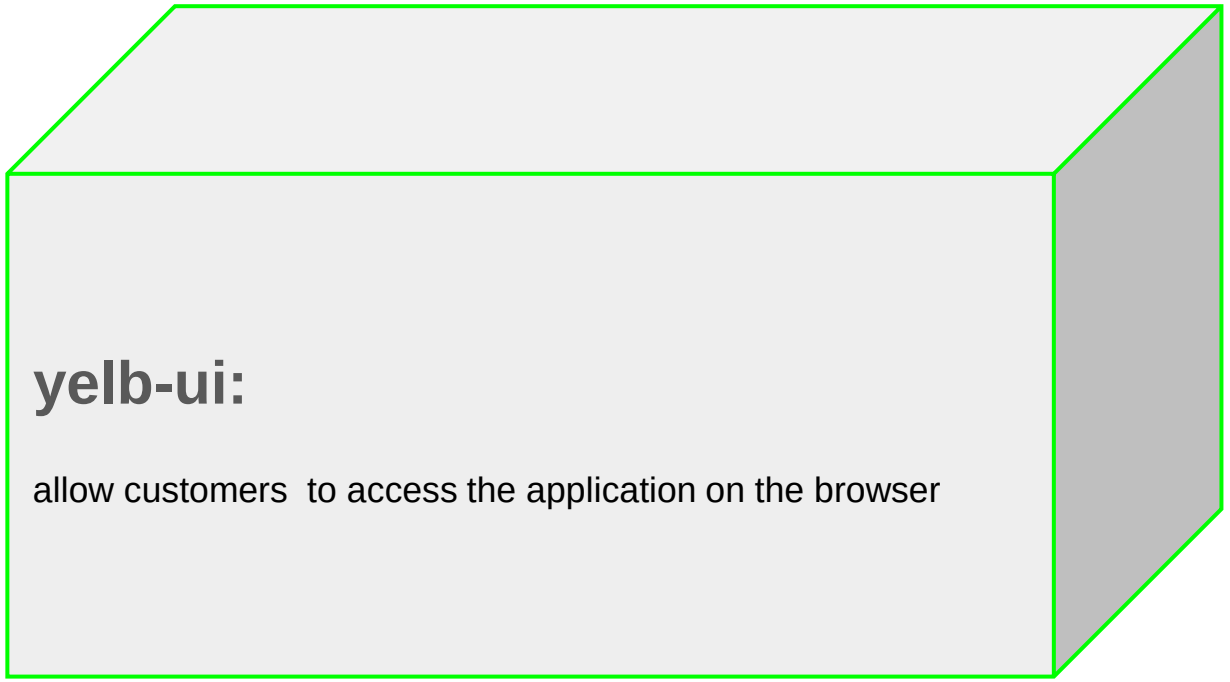
1- yelb-ui
2- yelb-appserver
3- redis-server
4- odilia-redis01
5- odilia-redis02
6- odilia-redis-sentinel01
7- odilia-redis-sentinel02
8- odilia-redis-sentinel03
9- yelb-db
10- odilia-db-replication01
11- odilia-db-replication02
12- odilia-db-replication03

Microservices

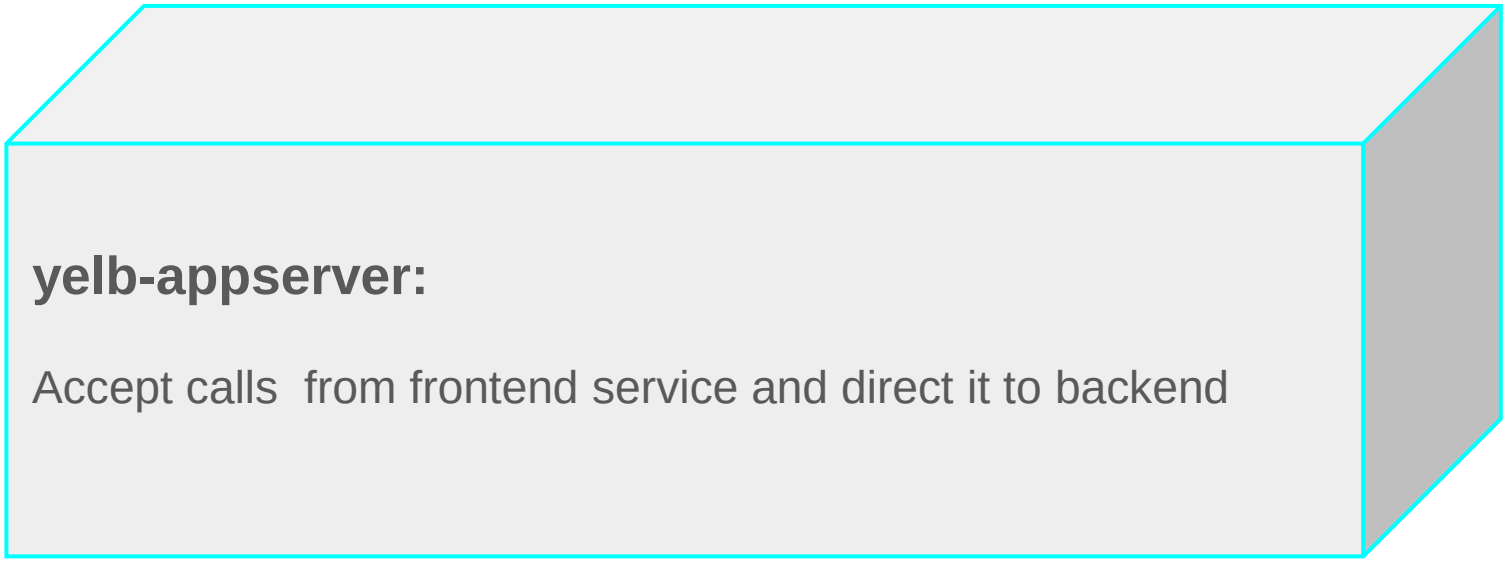


DEVOPS EASY LEARNING

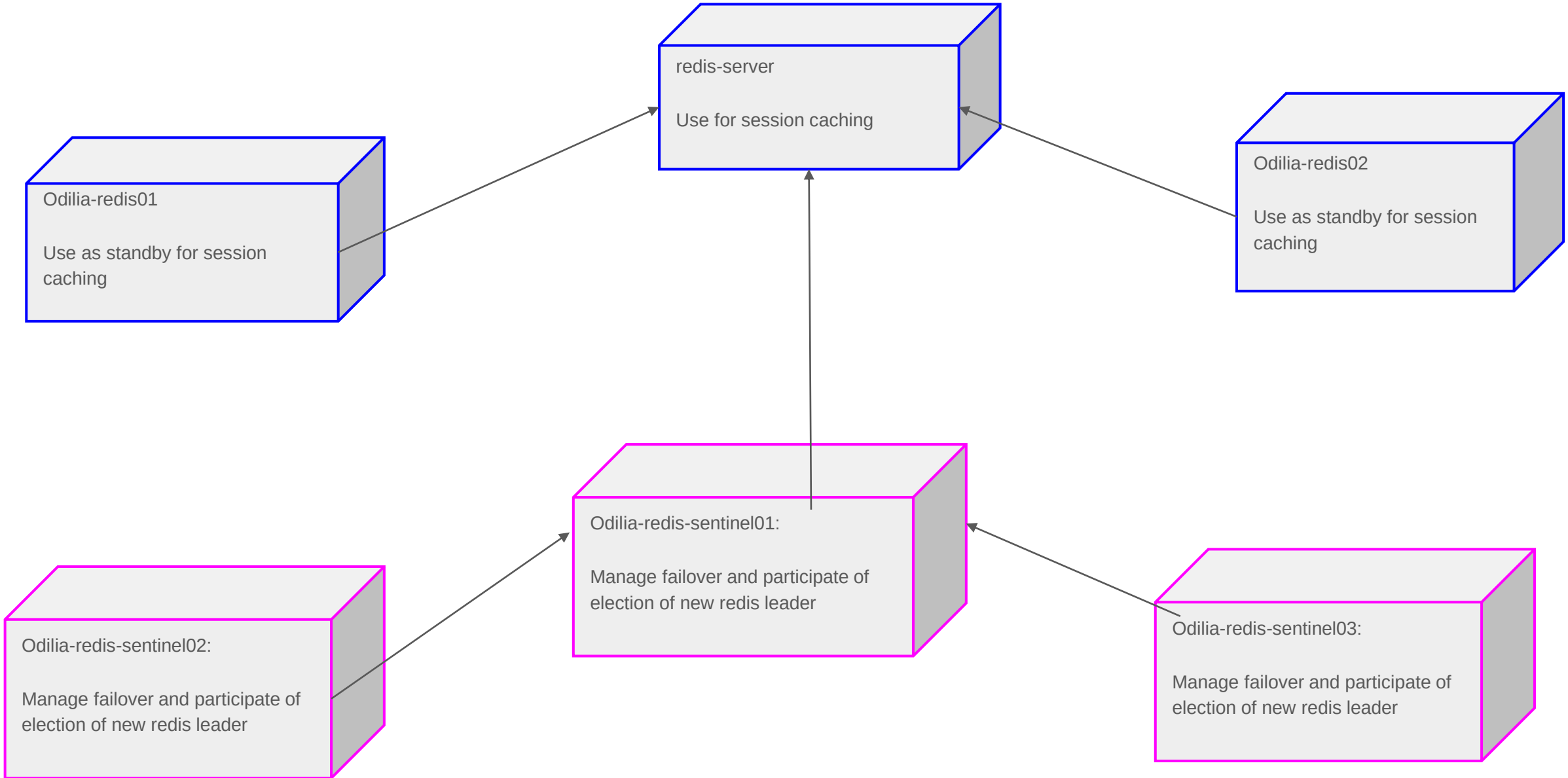
frontend



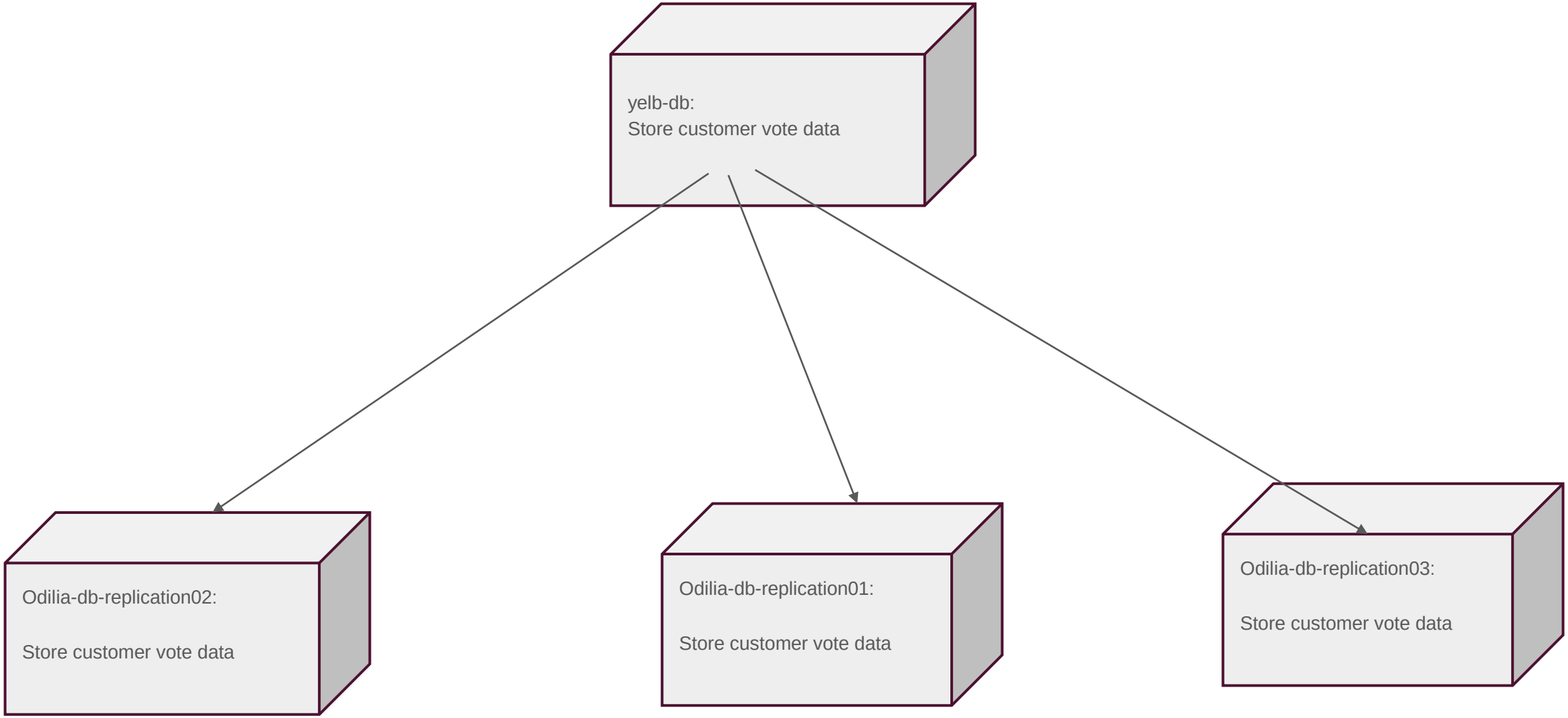
API

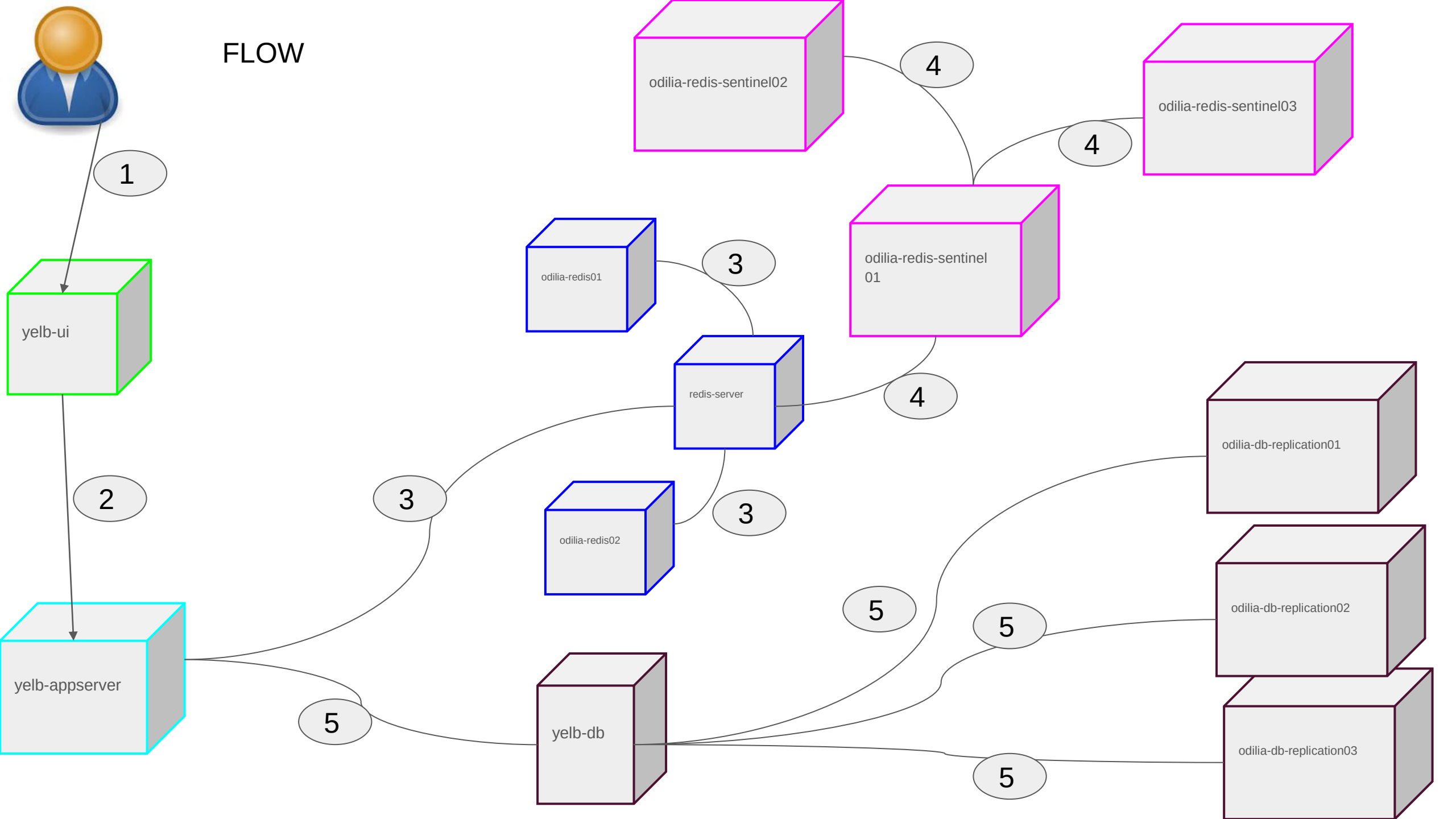


Backend


redis-server
Use for session caching
Odilia-redis01
Use as standby for session caching
Odilia-redis02
Use as standby for session caching
Odilia-redis-sentinel01:
Manage failover and participate of election of new redis leader
Odilia-redis-sentinel02:
Manage failover and participate of election of new redis leader
Odilia-redis-sentinel03:
Manage failover and participate of election of new redis leader

Backend


yelb-db:
Store customer vote data
Odilia-db-replication02:
Store customer vote data
Odilia-db-replication01:
Store customer vote data
Odilia-db-replication03:
Store customer vote data


FLOW
odilia-redis-sentinel02
odilia-redis-sentinel03
odilia-redis-sentinel
01
odilia-redis01
redis-server
odilia-redis02
yelb-ui
yelb-appserver
yelb-db
odilia-db-replication01
odilia-db-replication02
odilia-db-replication03
1
2
3
3
3
4
4
4
5
5
5
5

Images

mreferre/**yelb-ui**:0.7
mreferre/**yelb-appserver**:0.5
mreferre/**yelb-db**:0.5
redis:4.0.2

Create files and mount under

/etc/redis/redis.conf
inside the container

File name: redis-server

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

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

File name: odilia-redis02 & odilia-redis01

```
protected-mode no
port 6379
slaveof odilia-redis01 6379

protected-mode no
port 6379
slaveof redis-0 6379

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

All redis and sentinels use same docker image

Create files and mount under

/etc/redis/sentinel.conf
inside the container

Create files Odilia-redis-sentinel01 Odilia-redis-sentinel02 Odilia-redis-sentinel03
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
```

All redis and sentinels use same docker image

DEVOPS EASY LEARNING

Special instructions

All volumes driver should be set to **local**

All networks driver should be set to **bridge**

yeld-ui is the frontend service and exposed interbally at 80

Start redis replication with following command
redis-server /etc/redis/redis.conf

Start sentinel with following command

redis-sentinel /etc/redis/sentinel.conf

DEVOPS EASY LEARNING