DOCKER COMPOSE

DOCKER COMPOSE

Compose is a tool for defining and running multi-container Docker applications. With Compose, you use a YAML file to configure your application's services. Then, with a single command, you create and start all the services from your configuration.

It has commands for managing the whole **lifecycle** of your application:

- Start, stop, and rebuild services
- View the status of running services
- Stream the log output of running services
- Run a one-off command on a service

The key features of Compose that make it effective are:

- Have multiple isolated environments on a single host
- Preserve volume data when containers are created
- Only recreate containers that have changed
- Support variables and moving a composition between environments

DOCKER COMPOSE

- Basic operations: Build/push/run with docker compose
- Environment variables
- Volumes
- Network
- Orchestration tools depends_on / healthcheck / restart

Follow the example at https://github.com/ynov-campus-sophia/devops-B3-2023/docker/compose

BASIC OPERATIONS COMPOSE

```
# docker-compose.yaml
version: "3"
services:
  datacollect:
    build:
    context: ./datacollect-inov
  ports:
    - 5000:5000
  user: '1000'
```

docker-compose build	← build the image
docker-compose push	← push the image to registry
docker-compose up	← run the containers
docker-compose -f mydocker-compose.yaml up	← run the containers (specify file)
docker-compose up -d	← run the containers demonize mode
docker-compose down	← stop the containers

ENV VARIABLES COMPOSE

```
# docker-compose.yaml
version: "3"
services:
  pg:
    image: timescale/timescaledb-postgis:latest-pg13
    ports:
      - 5432:5432
    volumes:
      - pgdata:/var/lib/postgresql/data
      - ./postgres/sql:/docker-entrypoint-initdb.d/
    environment:
      POSTGRES_PASSWORD: mypassword              ←  INLINE ENV VAR
      POSTGRES_DB: mydb
      POSTGRES_USER: $POSTGRES_USER
```

- Otherwise create an env file at the same level of docker-compose file then use it in docker-compose
  `POSTGRES_USER=test`
- `All env var defined at system level (es. In the .bashrc) have higher priority`

VOLUMES COMPOSE

```
# docker-compose.yaml
version: "3"
services:
  pg:
    image: timescale/timescaledb-postgis:latest-pg13
    ports:
      - 5432:5432
    volumes:
      - pgdata:/var/lib/postgresql/data                ← NAMED VOLUMES
      - ./postgres/sql:/docker-entrypoint-initdb.d/    ← HOST MOUNTED
volumes:
  pgvol:
  pgdata:
```

- The most common volume declaration is done via **named volumes** hand **host mounted** volumes
- A volume can be provisioned via command line and then attached to a service after having defined it as external

```
docker volume create myvolume
```

```
volumes:
  myvolume:   ← add this in volumes section
    external: true
```

https://docs.docker.com/engine/reference/commandline/volume_create/

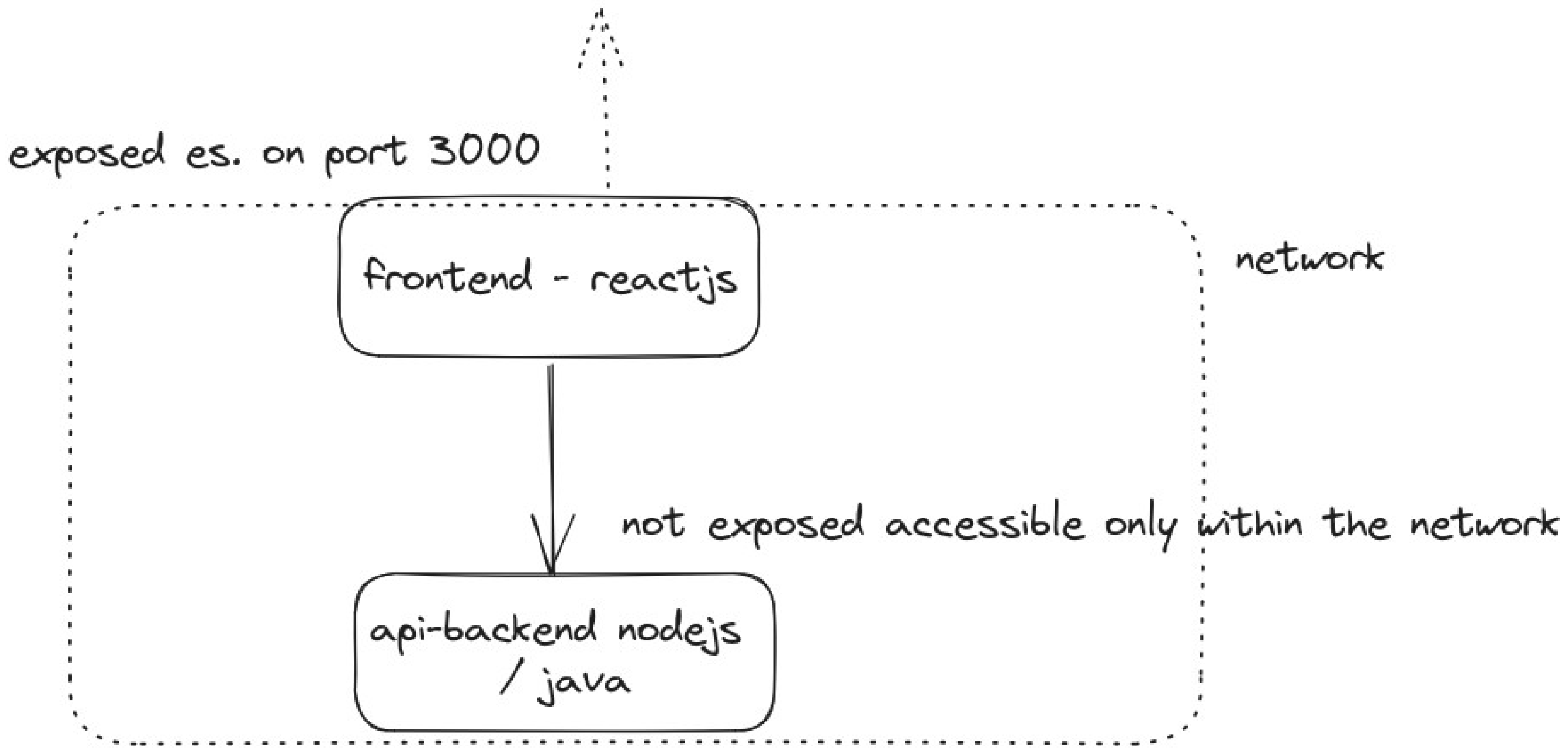
NETWORK COMPOSE

```
# docker-compose.yaml
version: "3"
services:
  pg:
    image: timescale/timescaledb-postgis:latest-pg13
    ports:
      - 5432:5432
    volumes:
      - pgdata:/var/lib/postgresql/data              ← NAMED VOLUMES
      - ./postgres/sql:/docker-entrypoint-initdb.d/  ← HOST MOUNTED
volumes:
  pgvol:
  pgdata:
```

https://stackoverflow.com/questions/48076605/how-to-configure-nginx-with-multiple-docker-compose-yml-files

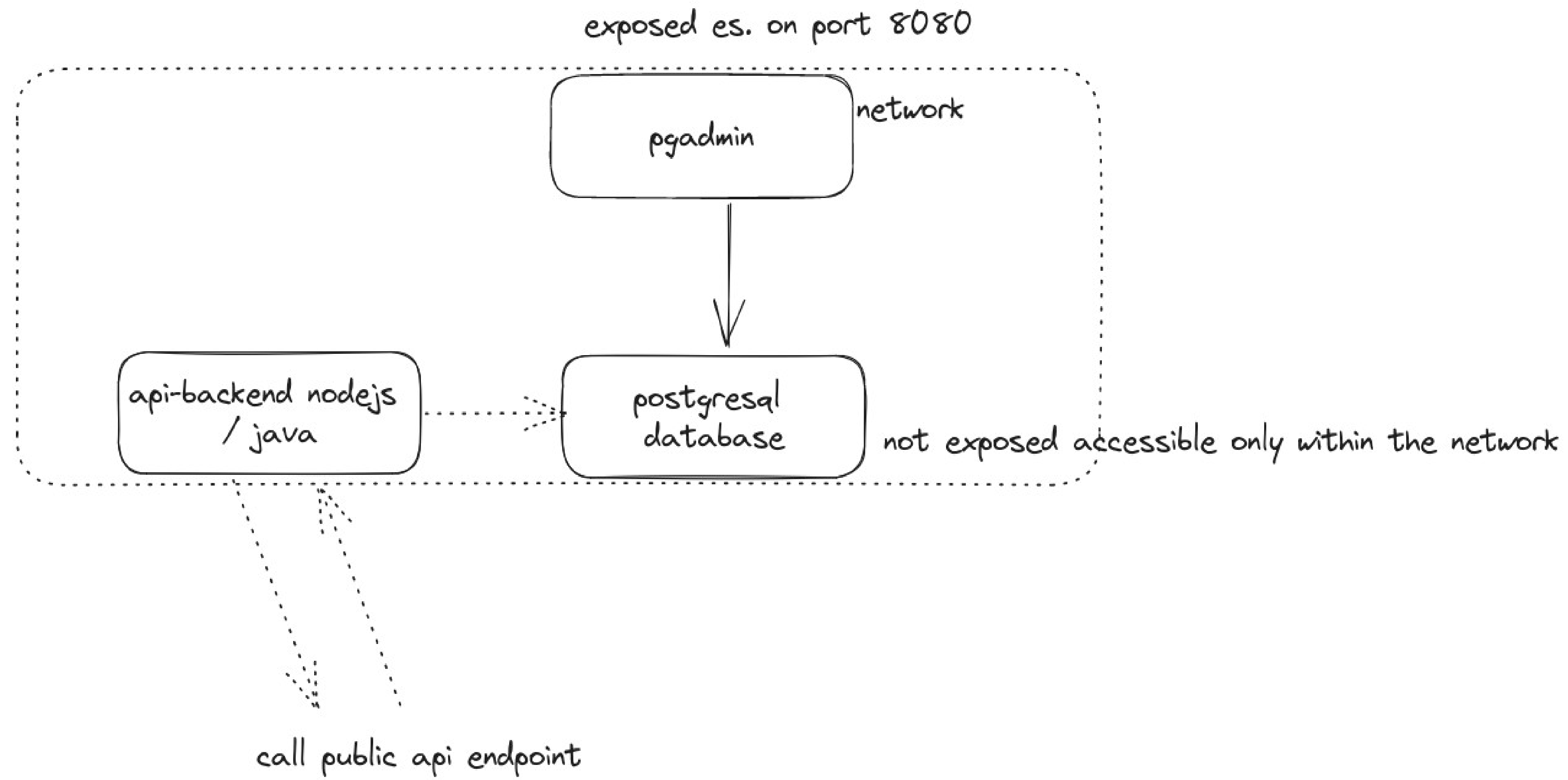
ARCHITECTURE MICROSERVICES

- Assignement1: deploy a react front that use data fetched from an api


exposed es. on port 3000
network
frontend - reactjs
not exposed accessible only within the network
api-backend nodejs / java

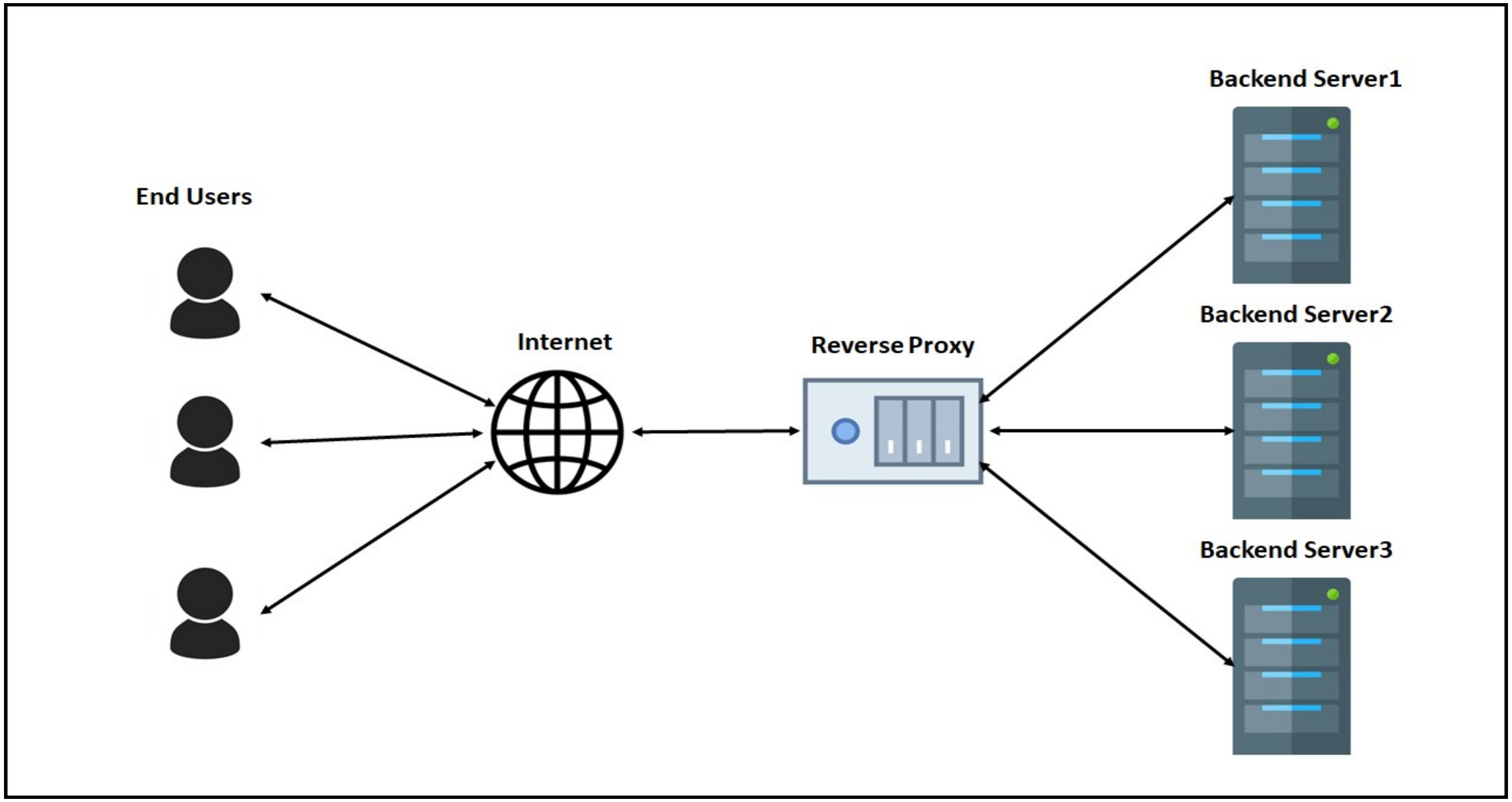
ARCHITECTURE MICROSERVICES

- Assignement2: Fetch data from crypto exchange api Kraken, collect data in postgres db



ARCHITECTURE MICROSERVICES

- Assignement3: Deploy a reverse proxy (nginx) to dispatch traffic to containers. Microservice architecture needs often a similar middleware component (like Kong or traefik).


End Users
Internet
Reverse Proxy
Backend Server1
Backend Server2
Backend Server3