Compose File

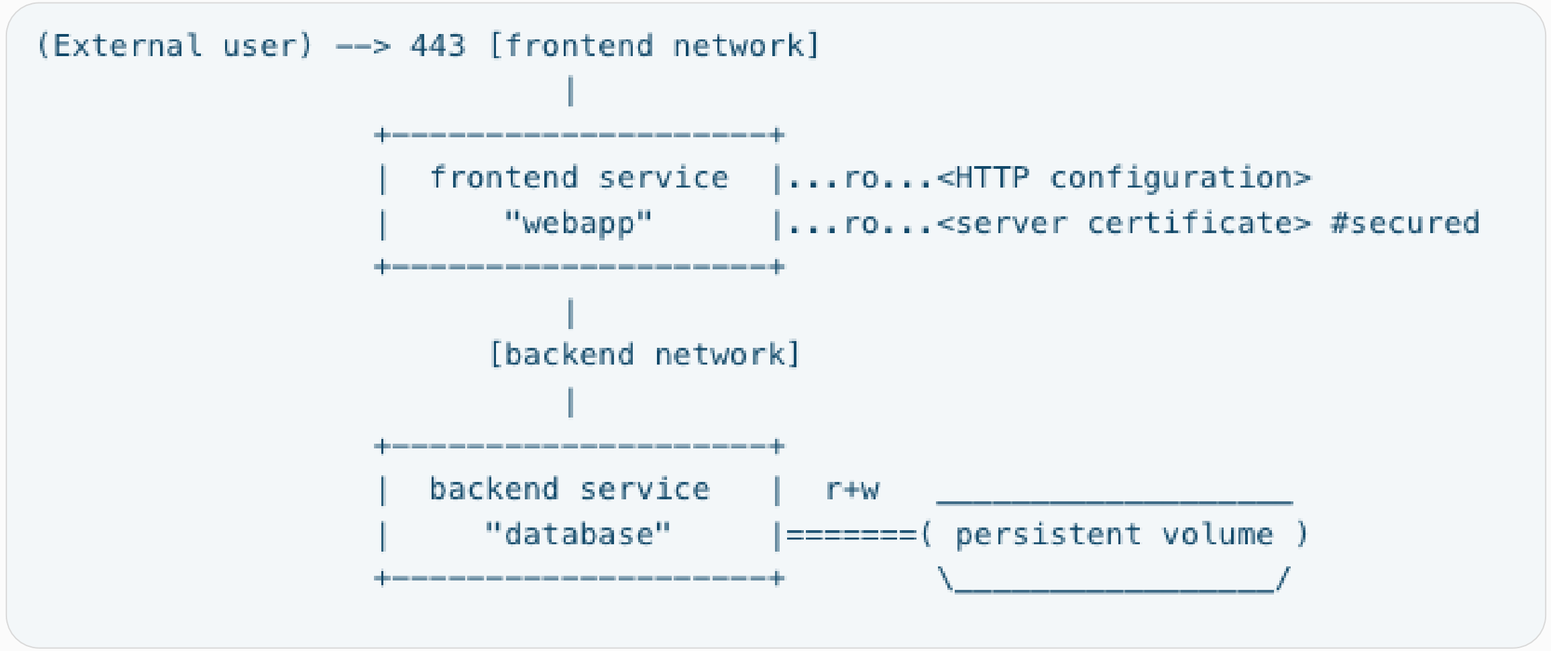
Цели

- Понять структуру файла compose.yaml
- Services, Networks, Volumes etc.
- docker compose up
  docker compose ps
  docker compose logs

Compose.yaml

- compose.yaml
- compose.yml
- docker-compose.yml
- docker-compose.yaml

Compose. Концепция



Изображение: Концепция Docker Compose

Compose File. Пример

```yaml
services:
  frontend:
    image: awesome/webapp
    ports:
      - "443:8043"
    networks:
      - front-tier
      - back-tier
    configs:
      - httpd-config
    secrets:
      - server-certificate

  backend:
    image: awesome/database
    volumes:
      - db-data:/etc/data
    networks:
      - back-tier
```

```yaml
volumes:
  db-data:
    driver: flocker
    driver_opts:
      size: "10GiB"

configs:
  httpd-config:
    external: true

secrets:
  server-certificate:
    external: true

networks:
  # The presence of these objects is sufficient to define them
  front-tier: {}
  back-tier: {}
```

Скриншоты спикера курса

Compose. Profiles

```yaml
services:
  foo:
    image: foo
  bar:
    image: bar
    profiles:
      - test
  baz:
    image: baz
    depends_on:
      - bar
    profiles:
      - test
  zot:
    image: zot
    depends_on:
      - bar
    profiles:
      - debug
```

```
$ docker compose --profile test up
$ COMPOSE_PROFILES=test docker compose up
$
$ docker compose --profile test --profile debug up
$ COMPOSE_PROFILES=test,debug docker compose up
$
$ # profile test activated
$ docker compose up bar
$
$ # profile test and service bar
$ docker compose up baz
$
$ # invalid model
$ docker compose up zot
$
$ # zot bar and baz activated
$ docker compose --profile test up zot
```

Скриншоты спикера курса

Compose. Version top-level element

```
version: "3.1"

services:
  reverse_proxy:
    build: ./reverse_proxy
    user: nginx

  database:
    build:
      context: ./database
```

Скриншот спикера курса

Compose. Service

```
services:
  wordpress:
    image: wordpress:latest
    volumes:
      - wp_data:/var/www/html
    ports:
      - 80:80
    restart: always
    environment:
      - WORDPRESS_DB_HOST=db
      - WORDPRESS_DB_USER=wordpress
      - WORDPRESS_DB_PASSWORD=wordpress
      - WORDPRESS_DB_NAME=wordpress
```

```
appserver:
  build:
    context: app
    dockerfile: Dockerfile
  image: atsea_app
  user: gordon
  ports:
    - "8080:8080"
    - "5005:5005"
  networks:
    - front-tier
    - back-tier
  secrets:
    - postgres_password
```

Compose. Environment

Map syntax:

```yaml
environment:
  RACK_ENV: development
  SHOW: "true"
  USER_INPUT:
```

Array syntax:

```yaml
environment:
  - RACK_ENV=development
  - SHOW=true
  - USER_INPUT
```

Compose. File .env

```
$ head .env
### NGINX #################################################

NGINX_SSL_PATH=./nginx/ssl/
######### TO-DO APP CONFIG ##############
NGINX_TODO_HOST_HTTP_PORT=8081
NGINX_TODO_HOST_HTTPS_PORT=4444
NGINX_TODO_SITE_PATH=./nginx/sites/todo.conf
NGINX_TODO_HOST_LOG_PATH=./../logs/nginx_todo
APP_CODE_TODO_PATH_HOST=./../todo
APP_CODE_TODO_PATH_CONTAINER=/var/www/todo


$ head docker-compose.yml
version: '2'

services:
   env_file: .env
   nginx_todo:
     build:
       context: ./nginx
     volumes:
       - ${APP_CODE_TODO_PATH_HOST}:${APP_CODE_TODO_PATH_CONTAINER}
       - ${NGINX_TODO_HOST_LOG_PATH}:/var/log/nginx
       - ${NGINX_TODO_SITE_PATH}:/etc/nginx/sites-available/todo.conf
```

Скриншот спикера курса

Compose. Networks

```yaml
services:
  frontend:
    image: awesome/webapp
    networks:
      - front-tier
      - back-tier

networks:
  front-tier:
  back-tier:
```

```yaml
networks:
  outside:
    external: true
```

Compose. Volumes

```yaml
services:
  backend:
    image: awesome/backend
    volumes:
      - type: volume
        source: db-data
        target: /data
        volume:
          nocopy: true
      - type: bind
        source: /var/run/postgres/postgres.sock
        target: /var/run/postgres/postgres.sock

volumes:
  db-data:
```

```yaml
services:
  backend:
    image: awesome/database
    volumes:
      - db-data:/etc/data

volumes:
  db-data:
    external: true
```

```yaml
volumes:
  example:
    driver_opts:
      type: "nfs"
      o: "addr=10.40.0.199,nolock,soft,rw"
      device: ":/docker/example"
```

Скриншоты спикера курса

Compose. Secrets

```yaml
services:
  frontend:
    image: awesome/webapp
    secrets:
      - server-certificate
secrets:
  server-certificate:
    file: ./server.cert
```

```yaml
secrets:
  server-certificate:
    external: true
```

Скриншоты спикера курса

Compose. Healthchecks

```yaml
healthcheck:
  test: ["CMD", "curl", "-f", "http://localhost"]
  interval: 1m30s
  timeout: 10s
  retries: 3
  start_period: 40s
```

```yaml
services:
  web:
    build: .
    depends_on:
      db:
        condition: service_healthy
      redis:
        condition: service_started
  redis:
    image: redis
  db:
    image: postgres
```

Итоги

- Поняли структуру файла compose.yaml
- Services, Networks, Volumes etc.
- docker compose up
  docker compose ps
  docker compose logs