Examples of LCARS-C deployment

A1: Local deployment with R/R Studio

Installation

Install R and R Studio. Clone this repository, open R Studio, and install LCARS-C from the repository's base folder including the lcarsM dependency:

```
devtools::install_local("dependencies/lcarsM.tar.gz")
devtools::install_local()
```

Example

```
library(lcarsc)
lcarsc::run_app(ecrf_database_driver = RSQLite::SQLite(),
        ecrf_dbhost = "dbtest",
        ecrf_dbname = "db_test_data.sqlite3",
        config_database_driver = RSQLite::SQLite(),
        config_dbhost = "dbtest",
        config_dbname = "db_test_cfg.sqlite3",
        options = list(host = '0.0.0.0', port = 3838))
```

A2: Local deployment using ShinyProxy and MariaDB with Docker

This can be used as a template for local deployments.

- Install the Docker engine
- Clone this repository and set the repository's folder as working directory.
- Create folders for database permanent storage:

```
mkdir -p ./shinyproxy/mariadb/logs
mkdir -p ./shinyproxy/mariadb/data
```

- Start the shinyproxy using docker-compose (the docker images are pulled automatically):

```
export DOCKERID=$(getent group docker | cut -d: -f3)
docker-compose -f ./shinyproxy/shinyproxy_local.yml up
```

- Open link to access shiny proxy (use test_user as username and password; see application file):
  http://localhost:7070/login

Backup and restore MariaDB database

- Backup:

```
docker exec CONTAINERID /usr/bin/mysqldump -u root --password=coucoutest mydbtest
> backup.sql
```

- Restore:

```
cat backup.sql | docker exec -i CONTAINERID /usr/bin/mysql -u root --
password=coucoutest mydbtest
```

B: Web deployment with Docker swarm

This deployment strategy is based on this tutorial.

Setup server

- Install Docker and join user into Docker group

```
sudo apt install docker.io
sudo systemctl enable --now docker
sudo usermod -aG docker [USER NAME]
```

- Re-login
- Pull Docker images

```
docker pull hstubbe/lcarsc:latest
docker pull mariadb
docker pull traefik
docker pull openanalytics/shinyproxy
docker pull quay.io/keycloak/keycloak
```

- Setup Docker swarm

```
docker swarm init
```

- Get tokens

```
docker swarm join-token worker
docker swarm join-token manager
```

- Create Docker networks

```
docker network create --driver=overlay sp-net
docker network create --driver=overlay traefik-public
```

- Setup traefik

```
export NODE_ID=$(docker info -f '{{.Swarm.NodeID}}')
export EMAIL=[VALID E-MAIL]
export DOMAIN=traefik.[FQDN]
export USERNAME=admin
export HASHED_PASSWORD=$(openssl passwd -apr1)
docker node update --label-add traefik-public.traefik-public-certificates=true
$NODE_ID

curl -L dockerswarm.rocks/traefik.yml -o traefik.yml
docker stack deploy -c traefik.yml traefik
docker stack ps traefik
docker service logs traefik_traefik
```

- Get this repository

```
git clone https://github.com/hcstubbe/lcars_webhosting.git
```

- Setup database

```
sudo mkdir -p /data/study/mariadb/ecrf/data/ /data/study/mariadb/ecrf/logs/
docker stack deploy -c shinyproxy/mariadb.yml db
```

- Updated keycloak theme (if changed)

```
docker build -t keycloak:updated_theme keycloak/.
```

- Deploy Keycloak

```
export KEYCLOAK_DOMAIN=keycloak.[FQDN]
docker stack deploy -c shinyproxy/keycloak.yml keycloak

export APP_DOMAIN=study.[FQDN]
```

```
export DOCKERID=$(getent group docker | cut -d: -f3)
docker stack deploy -c shinyproxy/shinyproxy.yml shinyproxy
```

- Deploy ShinyProxy

```
export APP_DOMAIN=study.[FQDN]
export DOCKERID=$(getent group docker | cut -d: -f3)
docker stack deploy -c shinyproxy/shinyproxy.yml shinyproxy
```

Backup database

- Create backup

```
docker exec CONTAINERID /usr/bin/mysqldump -u root --password=[YOUR PASSWORD] db >
backup.sql
```

- Restore

```
cat backup.sql | docker exec -i CONTAINERID /usr/bin/mysql -u root --password=
[YOUR PASSWORD] db
```

- Copy backup to local machine

```
scp [user]@[IP]:~/servername/backup.sql ~/backup.sql
```

Remove stacks

If you wish to stop the server use the following:

```
docker stack remove shinyproxy
docker stack remove keycloak
docker stack remove traefik
docker stack remove mariadb
docker network prune
```