opiasec Labs

opiasec Labs is a learning platform for application security aimed at developers and IT professionals/enthusiasts.

Getting Started (Back-end and Front-end)

Stack

- **Back-end**: Go, PostgreSQL, Redis
- **Front-end**: Angular, TypeScript,

Prerequisites

- Docker
- Docker Compose
- Go (for back-end development)
- NPM (for front-end development)

Installation

1. Clone the repository:

```
git clone https://github.com/AppSec-Digital/appseclabsPlataform.git
```

2. Navigate to the project directory:

```
cd appseclabsPlataform
```

3. Install dependencies:
   - For the back-end:

```
cd backend
go mod download
```

   - For the front-end:

```
cd frontend
npm install
```

4. Set up environment variables (back-end): Create a `.env` file in the back-end `deployment` directory and add the necessary environment variables (see below).
5. Start the application:
    - For the back-end:

```
make install
```

    - For the front-end:

```
cd frontend
npm start
```

Environment Variables

```
JWKS_URL=https://valued-wildcat-17.clerk.accounts.dev/.well-
known/jwks.json
DB_POSTGRES_URL=postgres://postgres:postgres@localhost:5432/postgres?
sslmode=disable
LAB_CLUSTER_BASE_URL=http://localhost:8084
REDIS_ADDR=localhost:6379
REDIS_PASSWORD=
REDIS_DB=0
LAB_IDE_BASE_URL=http://localhost:8082
```

Usage

- Access the application at `http://localhost:4200`.
- The back-end API is available at `http://localhost:8080`.

Makefile Commands

- `make install`: Install dependencies and set up the back-end.
- `local-install`: Install dependencies for back-end for local development. Usually used in development environments for quick readiness setup.
- `local-stop`: Stop the local back-end server. Stop for local dependencies started with `local-install`.
- `migrate-up`: Run database migrations.
- `migrate-down`: Rollback database migrations.

Roadmap

- [ ] Validate user input in the back-end (Validator).
- [ ] Add some validations in back-end to ensure atomic lab creation (only one active lab per user).
- [ ] Implement user roles and permissions.

- [ ] Ranking system for labs.
- [ ] Adjust webhook route(s) to be only accessible by the Cluster API (different port).
- [ ] Add unit tests for the back-end.
- [ ] Add integration tests for the back-end.
- [ ] Add unit tests for the front-end.

Getting Started (API)

Prerequisites for local development:

- Docker
- Kubernetes Cluster (install Kind) -> https://kind.sigs.k8s.io/docs/user/quick-start/
- Install Kubectl
- Golang
- Makefile

Step by step

Follow the step by step guide for local development.

1. Start and Configure Cluster

Create the Cluster with the configurations

```
make -C kubernetes/ install
```

2. Set environment variables. The following variables are **REQUIRED**

```
KUBECONFIG_BASE64="$(cat kubeconfig.b64)"
MONGODB_URI="mongodb://admin:admin@localhost:27017/?authSource=admin"
MONGODB_DATABASE=appseclabs
PORT=8084
REDIS_ADDR="localhost:6379"
LAB_BASE_URL="http://localhost:8082"
DOCKER_PAT="ghp_{token}"
JWKS_URL="https://valued-wildcat-17.clerk.accounts.dev/.well-
known/jwks.json"
GITHUB_INIT_CONTAINER_USERNAME="username"
GITHUB_INIT_CONTAINER_PAT="github_pat_{}"
```

> KUBECONFIG_BASE64 is the variable that will contain the configuration (.kube/config) in base64.
> The makefile command can help (it will create the file 'kubeconfig.b64' from .kube/config):
>
> ```
> make get-kube-config
> ```

3. Upload the Laboratory images to the Cluster

For the laboratory upload to work correctly, the images of the laboratories need to be in the cluster. In Kind you can import the image into the cluster:

```
kind load docker-image ghcr.io/vitor-mauricio/lab-runner-example:dev
kind load docker-image ghcr.io/vitor-mauricio/lab-runner-copy:dev
```

Additionally, for the daemon (dind) of each laboratory to work (and in an optimized way), the images of the dynamic labs need to be in the local registry of the cluster:

```
make -C kubernetes/sync-images sync-images
```

> This step may take a while.

This command will synchronize the images of the labs with the local registry of the cluster. If you want to add a new lab, you need to add it to the `kubernetes/sync-images/sync-images.yaml` file.

4. Make install. Run Make install to start the local environment

To start the entire API environment in Docker, go to the `api` folder and run:

```
make install
```

or

To start only MongoDB and Redis via Docker:

```
make local-install
```


```
make local-install
```