

Download and Run qvaring container

Prerequisites.

AWS CLI

<https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html>

Docker

<https://docs.docker.com/engine/install/>

1 - Configure aws cli running

```
aws configure
```

And enter these details:

```
AWS Access key id: AKIA242OUIV5MZVE5BAD
AWS Secret Access key: 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
region: us-west-1
```

Check that the configuration is working with the command

```
$ aws sts get-caller-identity
```

It should return this:

```
{
  "UserId": "AIDA242OUIV5AYOHRXNBT",
  "Account": "749099828602",
  "Arn": "arn:aws:iam::749099828602:user/mindi"
}
```

2 - Select the image to download

Login to the aws management console through this link:

<https://749099828602.signin.aws.amazon.com/console>

user: mindiii

password: Fire@1231

Go to ECR (Elastic container registry) on the North california region ([link](#)) and select the repo qvaring-qva_staging ([link](#)).

Each image is tagged with the same id of the commit on the code repo. So select the one you want to download.

3 - Login to registry through the cli

```
aws ecr get-login-password --region us-west-1 | docker login --username AWS --password-stdin 749099828602.dkr.ecr.us-west-1.amazonaws.com
```

```
aws ecr get-login-password --region us-west-1 | docker login --username
AWS --password-stdin 749099828602.dkr.ecr.us-west-1.amazonaws.com
```

It will return:

```
WARNING! Your password will be stored unencrypted in /home/mijack/.
docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-
store
```

```
Login Succeeded
```

4 - pull the image to your local machine

Run a command like this:

```
749099828602.dkr.ecr.us-west-1.amazonaws.com/qvaring-qva_staging:
<COMMIT_ID>
```

like for example:

```
749099828602.dkr.ecr.us-west-1.amazonaws.com/qvaring-qva_staging:
87a2c5c0b813003def053fe2be0640ae188b73d9
```

Now there are two options.

1 - Run a bash shell on a new container in the context of the image so you can check the files inside

```
docker run -it --rm --name qvaring 749099828602.dkr.ecr.us-west-1.
amazonaws.com/qvaring-qva_staging:
87a2c5c0b813003def053fe2be0640ae188b73d9 bash
```

2 - Run the container and start the services

2.1 - Connect to the production vpn to get access to staging database

2.2 - create a directory env_file under your home directory

```
mkdir $HOME/env_file/
```

and put inside the .env file for testing (find it at the end of the document)

so the file should appear under this command

```
cat $HOME/env_file/.env
```

2.3 Run the container on a dedicated terminal. Don't close it. It will show the logs

```
docker run -it --rm -v /$HOME/env_file/:/root/env_file/ -p 8080:80 --
name qvaring 749099828602.dkr.ecr.us-west-1.amazonaws.com/qvaring-
qva_staging:<IMAGE_ID>
```

```
docker run -it --rm -v /$HOME/env_file/:/root/env_file/ -p 8080:80 --name qvaring 749099828602.dkr.ecr.us-
west-1.amazonaws.com/qvaring:<IMAGE_ID>
```

A real example:

```
docker run -it --rm -v /$HOME/env_file/:/root/env_file/ -p 8080:80 --
name qvaring 749099828602.dkr.ecr.us-west-1.amazonaws.com/qvaring-
qva_staging:87a2c5c0b813003def053fe2be0640ae188b73d9
```

Now you should be able to browse the site through <http://localhost:8080> on your browser.

environment (.env) file for staging: [here](#)