Deployment Instructions

Deploying the platform requires at least two instances, one frontend and one backend. Both requires docker and the backend requires docker-compose

Frontend:

Recommendation: use Vercel to deploy since it is a Nextjs project, otherwise here are the instructions to deploy the static using your own instance
Requirements:

- Docker
- GitHub actions running the Deploy static CI

Start steps:

1. Git clone the repository and change to the static-build branch
2. Change current directory to skillcity-platform
3. Assuming docker is installed and running with systemd:
    a. Build docker image with `docker image build -t static .`
    b. To start running the docker image, do `docker run -d -p 80:80 static` to run with http

Updating (To sync new changes):

1. List the running docker instances with `docker ps`

2. Stop the docker with the old version `docker stop <ID#HERE>` but replace <ID#HERE> with the docker running listed in the previous step
3. Remove the docker images with `docker rm <ID#HERE>`
4. Follow the starting steps to set up the docker with the new changes (git pull instead of git clone since the changes is all you need)

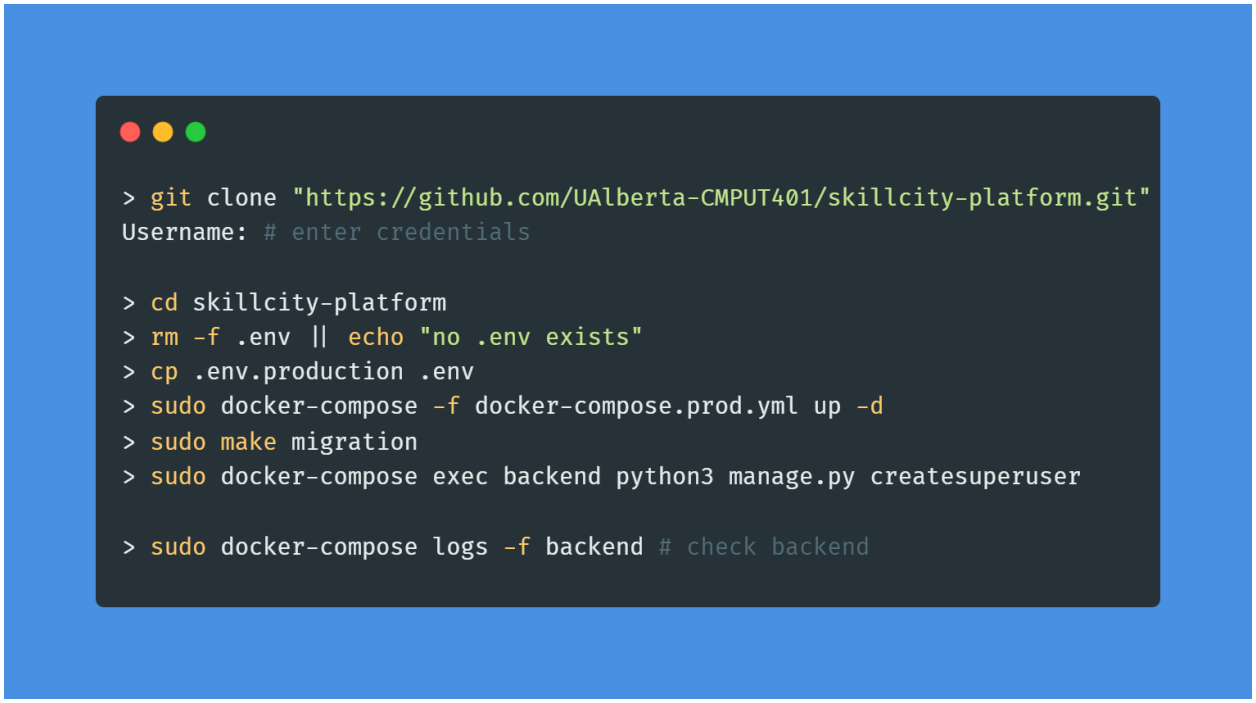
Backend:

Recommendation: Run this one an ec2 instance or another cloud instance. Alternatively do an elastic beanstalk deploy but those steps are not shown here so you will need to figure that out yourself.
Requirements:

- Docker
- Docker-compose
- Docker volume of a good size since it will store DB with images

Steps:

1. Connect the volume to the new instance by mounting it on (see https://wiki.cybera.ca/display/RAC/Rapid+Access+Cloud+Guide%3A+Part+1#RapidAccessCloudGuide:Part1-Volumes for reference)
2. Git clone the repository and change current directory to skillcity-platform
3. Use the .env.prod as the environment variables. This can be done multiple ways but one way is:
    a. `rm -f .env || echo “no .env exists”` if an .env exists
    b. `cp .env.production .env` to copy the production environment variables to current instance
4. Assuming Docker and docker-compose is installed do `sudo docker-compose -f docker-compose.prod.yml up -d`
5. Run db migrations with `sudo make migration`
6. Create an admin user
    a. Do `sudo docker-compose exec backend python3 manage.py createsuperuser` to create admin user
    b. `sudo make createsuperuser` for an admin account of user:admin, pwd:admin but this is **not recommended**.
7. Check if it is running with `sudo docker-compose logs -f backend`



```
> git clone "https://github.com/UAlberta-CMPUT401/skillcity-platform.git"
Username: # enter credentials

> cd skillcity-platform
> rm -f .env || echo "no .env exists"
> cp .env.production .env
> sudo docker-compose -f docker-compose.prod.yml up -d
> sudo make migration
> sudo docker-compose exec backend python3 manage.py createsuperuser

> sudo docker-compose logs -f backend # check backend
```

Updating (To sync new changes):

1. Git pull changes
2. Run db migrations with `sudo make migration`


> git pull
> sudo make migration