

Challenge 3

- Clone both repo from github
- we have to deploy the repos on the server using docker compose. So, I first write the docker files for both sites and then write a docker compose file

1) for saic website as it is based on html-css-js

```
1 FROM nginx:latest
2
3 COPY . /usr/share/nginx/html/
```

2) for github languages as it is based upon ruby on rails

```
1 FROM ruby:2.3.1-alpine
2 WORKDIR /app
3
4 RUN apk --no-cache add \
5     build-base \
6     libcurl \
7     curl-dev \
8     nodejs \
9     postgresql-dev \
10    tzdata
11
12 RUN gem install bundler -v '1.17.3'
13 COPY Gemfile Gemfile.lock /app/
14
15 RUN bundle install
16
17 COPY . /app
18
19 EXPOSE 3000
20
21 CMD ["rails", "server", "-b", "0.0.0.0"]
22
```

- Make a Docker compose file which contains info of both the website and it will be used to run both the containers.

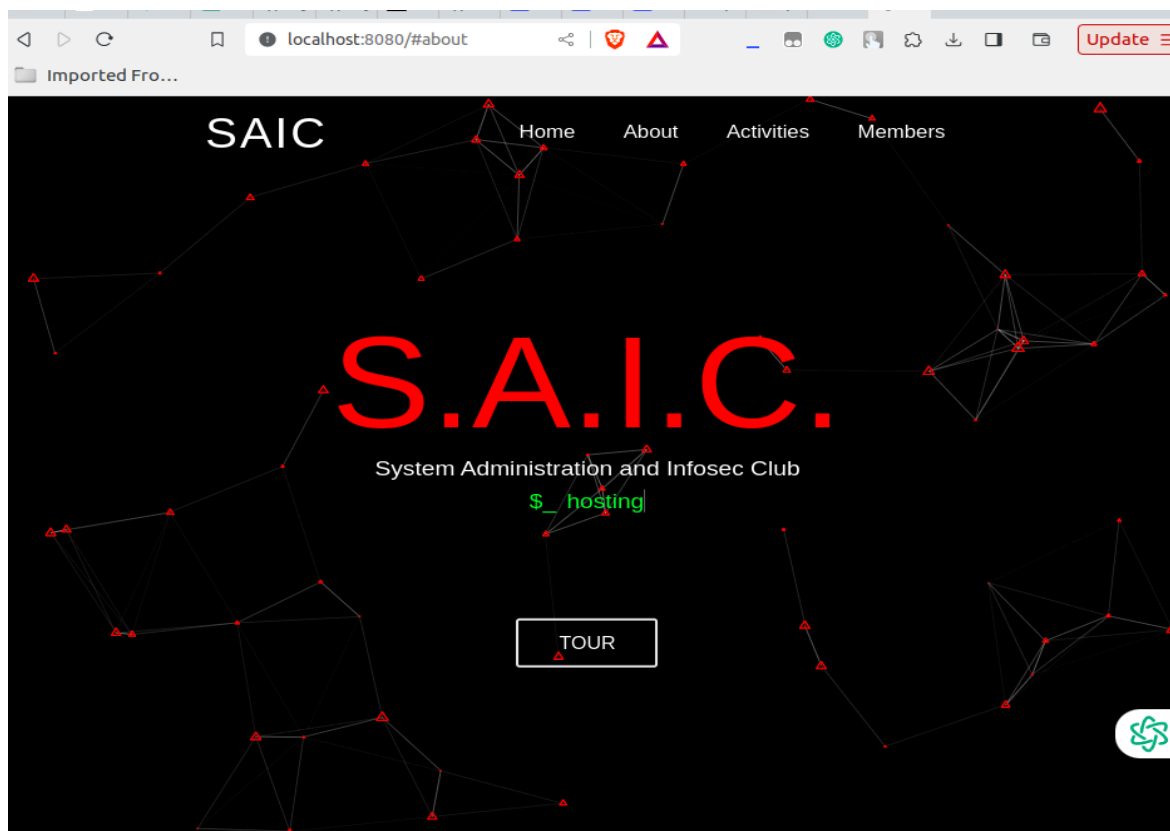
```
1 version: '3'
2
3 services:
4
5   postgres:
6     image: postgres:latest
7     volumes:
8       - /home/utsav/Desktop/challenge-3/postgres-data:/var/lib/postgresql/data
9     networks:
10      - ruby-on-rails-network
11     environment:
12       POSTGRES_PASSWORD: password
13       POSTGRES_DB: github-languages_development
14       POSTGRES_USER: github-languages
15
16   saic-website:
17     build: ./SAIC-Website
18     ports:
19       - "8080:80"
20     networks:
21       - html-css-js-network
22     volumes:
23       - /home/utsav/Desktop/challenge-3/saic-data:/app/data
24
25   github-lang:
26     build: ./github-languages-master
27     ports:
28       - "3000:3000"
29     networks:
30       - ruby-on-rails-network
31     depends_on:
32       - postgres
33     environment:
34       RAILS_ENV: development
35       DATABASE_HOST: postgres
36
37 networks:
38   html-css-js-network:
39     driver: bridge
40
41   ruby-on-rails-network:
42     driver: bridge
43
44
```

To run this file we have to go to the directory where our compose file is present in our terminal and then type :- `docker compose up`

Then it will install necessary dependencies which we have written in our docker files and run on the port mentioned in the compose file on our localhost.

[08]

```
utsav@utsav-Modern-14-B5M:~/Desktop$ cd challenge-3/
utsav@utsav-Modern-14-B5M:~/Desktop/challenge-3$ docker compose up
[+] Building 3.8s (21/21) FINISHED                                docker:default
=> [github-lang internal] load .dockerignore                    0.0s
=> == transferring context: 2B                                  0.0s
=> [github-lang internal] load build definition from Dockerfile 0.0s
=> == transferring dockerfile: 372B                             0.0s
=> [github-lang internal] load metadata for docker.io/library/ruby:2.3.1-alpine 3.1s
=> [saic-website internal] load .dockerignore                  0.0s
=> == transferring context: 2B                                  0.0s
=> [saic-website internal] load build definition from Dockerfile 0.0s
=> == transferring dockerfile: 89B                              0.0s
=> [saic-website internal] load metadata for docker.io/library/nginx:latest 3.8s
=> [saic-website auth] library/nginx:pull token for registry-1.docker.io 0.0s
=> [github-lang auth] library/ruby:pull token for registry-1.docker.io 0.0s
=> [github-lang 1/7] FROM docker.io/library/ruby:2.3.1-alpine@sha256:8d5ca285f1 0.0s
=> [github-lang internal] load build context                    0.0s
=> == transferring context: 6.75kB                              0.0s
=> CACHED [github-lang 2/7] WORKDIR /app                       0.0s
=> CACHED [github-lang 3/7] RUN apk --no-cache add build-base libcurl 0.0s
=> CACHED [github-lang 4/7] RUN gem install bundler -v '1.17.3' 0.0s
=> CACHED [github-lang 5/7] COPY Gemfile Gemfile.lock /app/    0.0s
=> CACHED [github-lang 6/7] RUN bundle install                 0.0s
=> [github-lang 7/7] COPY . /app                               0.0s
=> [github-lang] exporting to image                             0.0s
=> == exporting layers                                          0.0s
=> == writing image sha256:bbab0fc943b3433fb692183451fecc0ec56ee892530170823f88 0.0s
=> == naming to docker.io/library/challenge-3-github-lang      0.0s
=> [saic-website internal] load build context                  0.0s
=> == transferring context: 2.65kB                              0.0s
```



I am to run the saic website but got error while github -languages.

Error 1: this is the error when i first run the compose file .

```
postgres-1 | 2024-01-09 21:09:14.332 UTC [1] LOG: database system is ready to accept connections
github-lang-1 | could not connect to server: No such file or directory
github-lang-1 | Is the server running locally and accepting
github-lang-1 | connections on Unix domain socket "/tmp/.s.PGSQL.5432"?
github-lang-1 | Couldn't create database for {"adapter"=>"postgresql", "encoding"=>"unicode", "pool"=>5, "database"=>"github-languages_develo
github-lang-1 | rails aborted!
github-lang-1 | PG::ConnectionBad: could not connect to server: No such file or directory
github-lang-1 | Is the server running locally and accepting
github-lang-1 | connections on Unix domain socket "/tmp/.s.PGSQL.5432"?
github-lang-1 | /usr/local/bundle/gems/activerecord-5.0.0.1/lib/active_record/connection_adapters/postgresql_adapter.rb:671:in `initialize'
github-lang-1 | /usr/local/bundle/gems/activerecord-5.0.0.1/lib/active_record/connection_adapters/postgresql_adapter.rb:671:in `new'
```

Possible Reason according to me : i think i have mistaking in something like database.yml file that's why i am not able to connect it to postgresql but i tried to change many time something both in compose and database.yml files but still got the error

Error 2: error after making changes whose compose file i shown above

```
Puma caught this error: SCRAM authentication requires libpq version 10 or above
(PG::ConnectionBad)
/usr/local/bundle/gems/activerecord-5.0.0.1/lib/active_record/connection_adapters/postgresql_adapter.rb:671:in
`initialize'
```

Backup-script.sh is used to backup the data everyday at 11:55pm we can use cron jobs in linux and schedule task in windows here i use cron jobs

```
GNU nano 4.8 /tmp/crontab.1VWnP6/crontab Modified
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
55 23 * * * /home/utsav/Desktop/challenge-3/backup-script.sh

File Name to Write: /tmp/crontab.1VWnP6/crontab
^G Get Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel        M-M Mac Format  M-P Prepend    ^T To Files
```

To make the script executable

I ran :- `chmod +x /home/utsav/Desktop/challenge-3/backup-script.sh`

Ans2:-

Network Type Used : Bridge Network

Reason: Deploying both the `saic website` and `github languages` on a bridge network provides an optimal balance between isolation and inter-container communication within the same host environment. Bridge network allows the containers to communicate efficiently. This is particularly advantageous for securing the deployment on a single host. And here we are deploying both containers on same host and