DevOps: Dato Dzneladze

1. install ubuntu server
2. install docker, docker-compose, npm
3. clone repository: https://github.com/gwynbleidd0014/6thElement.git
4. make Dockerfile and docker-compose

```
version: '3.8'

services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    ports:
      - "8000:8080"
    depends_on:
      - db
    environment:
      -
ConnectionStrings__DefaultConnection=Server=db;Database=SixthElement;User=sa;Pass
word=6thElement;TrustServerCertificate=True;
      - ASPNETCORE_ENVIRONMENT=Development
    volumes:
      - ./server/6thElement/Images:/app/Images
    networks:
      - app-network

  db:
    image: mcr.microsoft.com/mssql/server:2019-latest
    environment:
      - ACCEPT_EULA=Y
      - SA_PASSWORD=6thElement
      - MSSQL_PID=Express
    ports:
      - "1433:1433"
    volumes:
      - sqldata:/var/opt/mssql
```

```yaml
    networks:
      - app-network

networks:
  app-network:
    driver: bridge

volumes:
  sqldata:
  images_volume:
```

Dokerfile

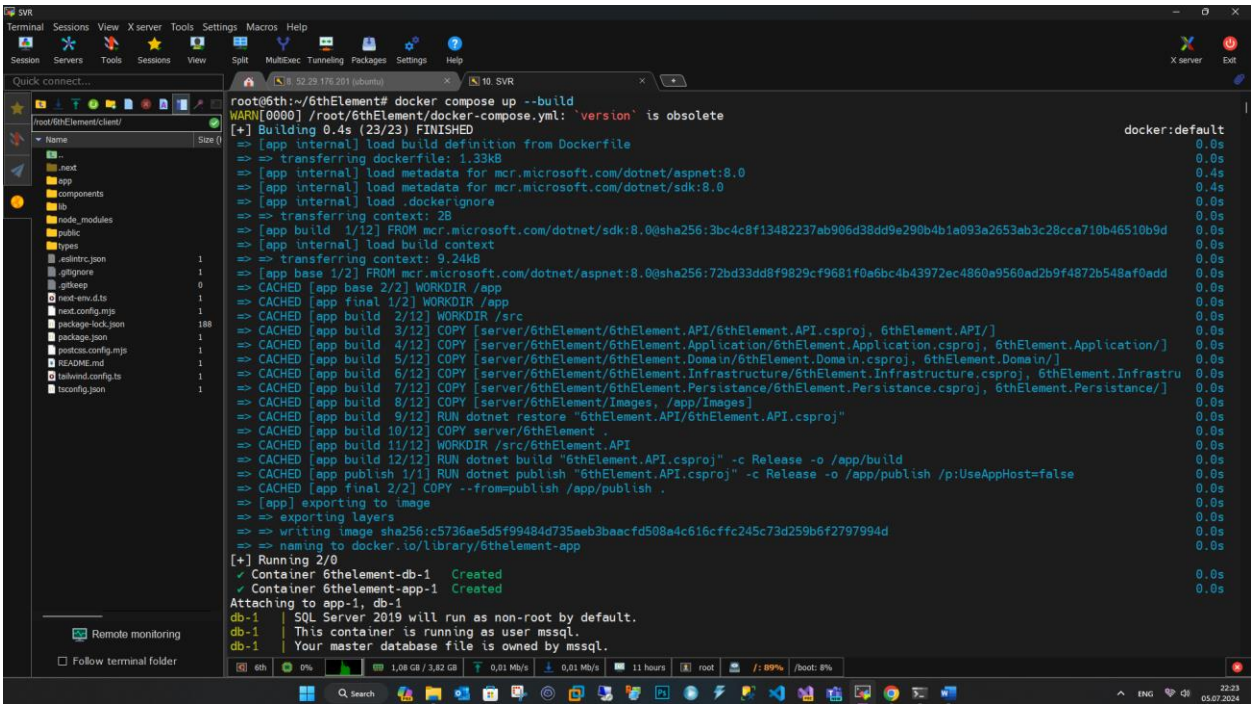
```dockerfile
# ბაზისური იმიჯი .NET 8.0-სთვის
FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
WORKDIR /app
EXPOSE 8080

# ბილდის იმიჯი
FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
WORKDIR /src
COPY ["server/6thElement/6thElement.API/6thElement.API.csproj",
"6thElement.API/"]
COPY ["server/6thElement/6thElement.Application/6thElement.Application.csproj",
"6thElement.Application/"]
COPY ["server/6thElement/6thElement.Domain/6thElement.Domain.csproj",
"6thElement.Domain/"]
COPY
["server/6thElement/6thElement.Infrastructure/6thElement.Infrastructure.csproj",
"6thElement.Infrastructure/"]
COPY ["server/6thElement/6thElement.Persistance/6thElement.Persistance.csproj",
"6thElement.Persistance/"]
COPY ["server/6thElement/Images", "/app/Images"]
RUN dotnet restore "6thElement.API/6thElement.API.csproj"
COPY server/6thElement .
WORKDIR "/src/6thElement.API"
RUN dotnet build "6thElement.API.csproj" -c Release -o /app/build

# პუბლიკაციის იმიჯი
FROM build AS publish
```

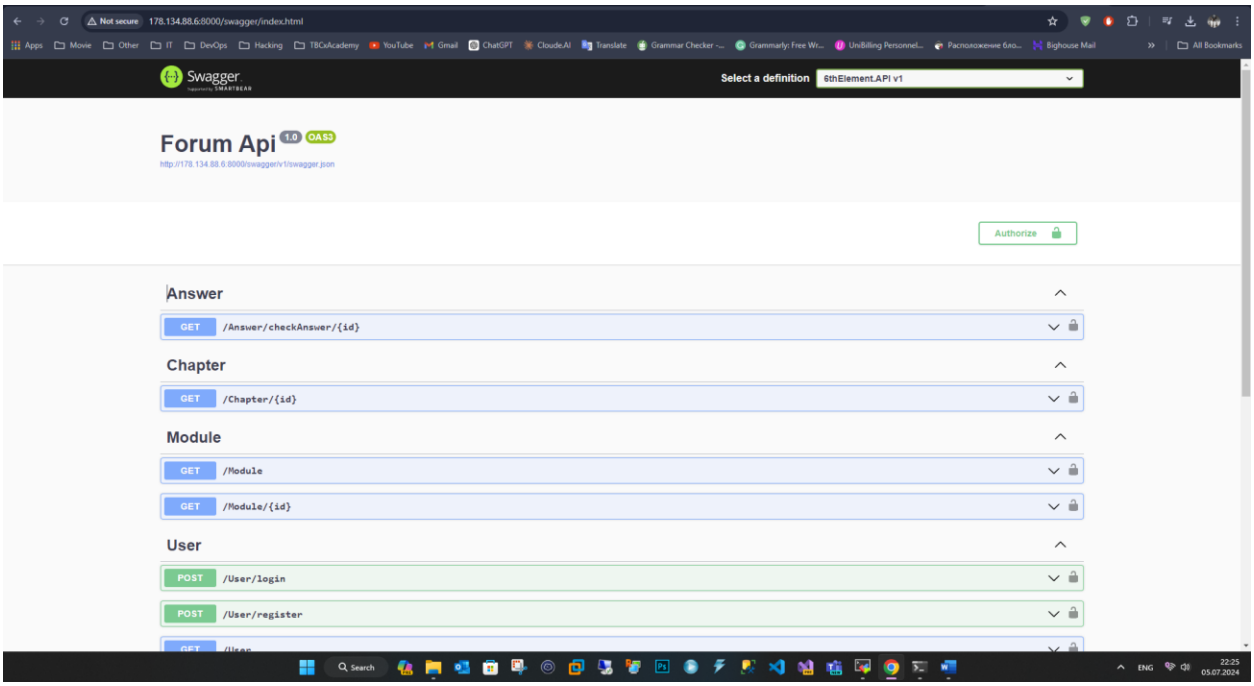
```
RUN dotnet publish "6thElement.API.csproj" -c Release -o /app/publish
/p:UseAppHost=false

# საბოლოო იმეჯი
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "6thElement.API.dll"]
```

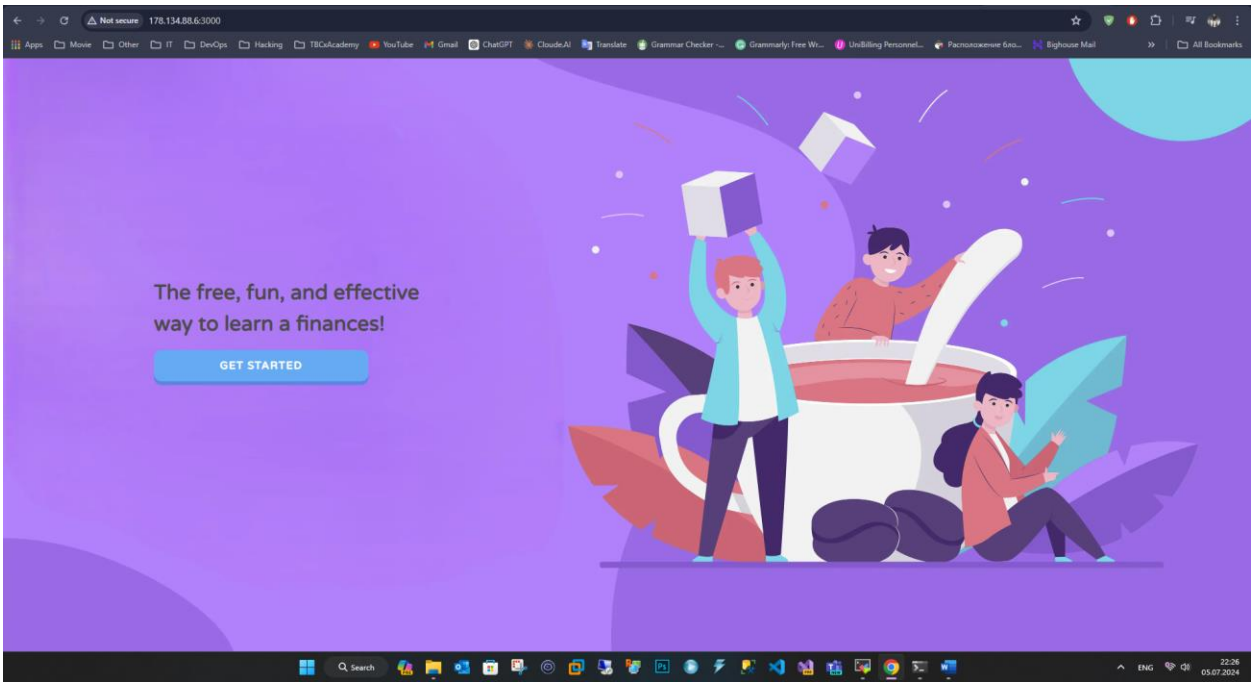


5. run npm build and npm start

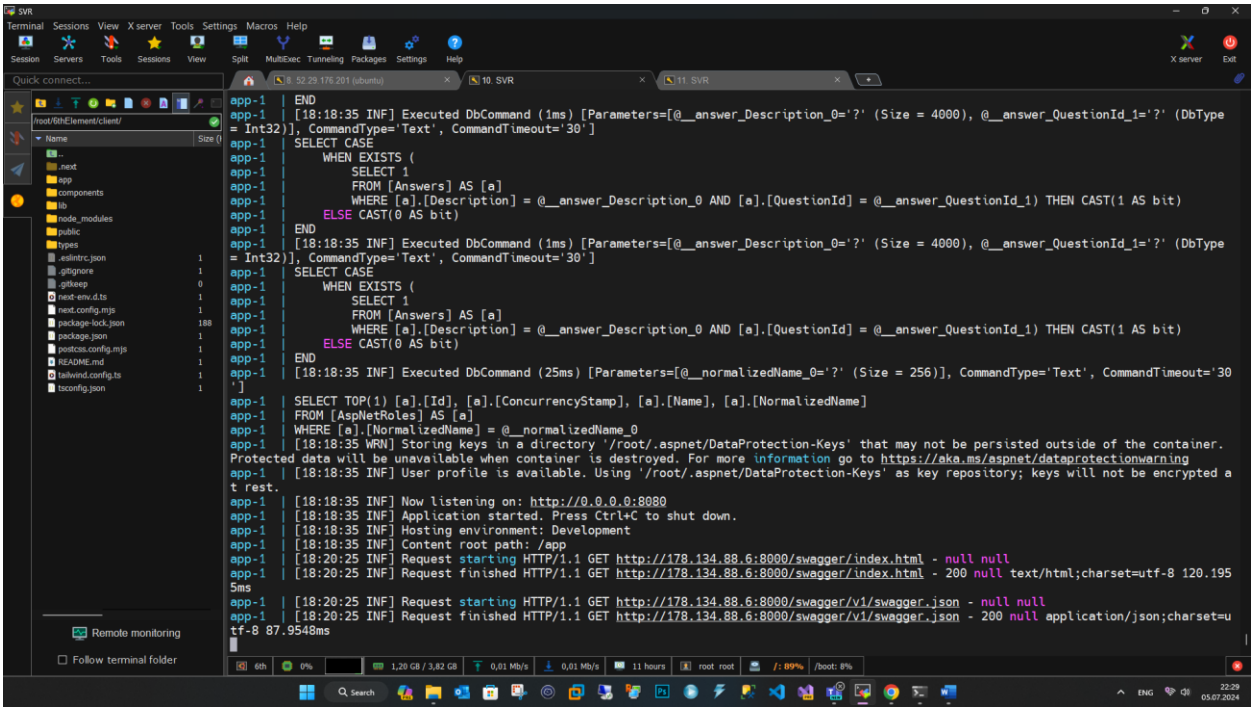
Runed API



Runed website



Igos



Install app on AWS

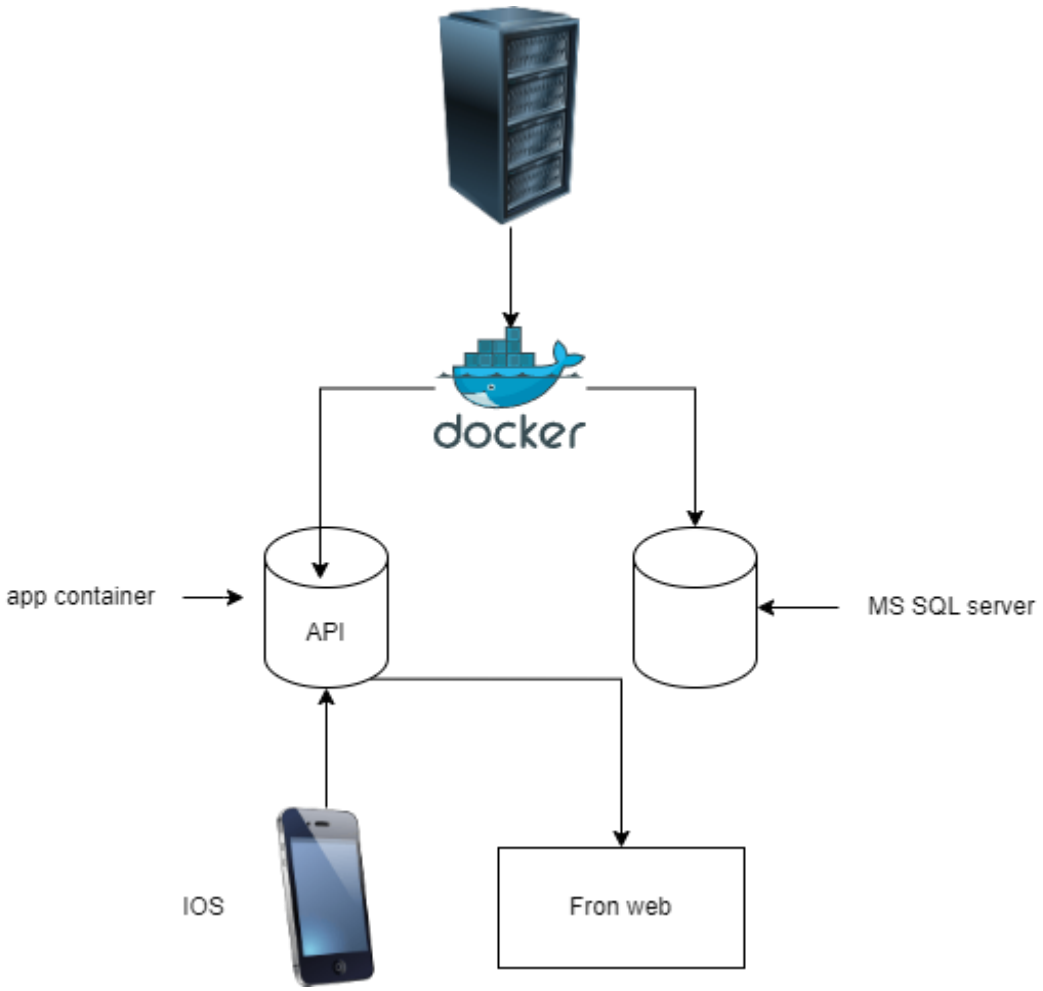
1. Create instance

2. run GitLab CI/CD

```
stages:
  - deploy

deploy:
  stage: deploy
  before_script:
    - apt-get update
    - apt-get install -y openssh-client
    - eval $(ssh-agent -s)
    - echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add -
    - mkdir -p ~/.ssh
    - chmod 700 ~/.ssh
    - echo "$SSH_KNOWN_HOSTS" > ~/.ssh/known_hosts
    - chmod 644 ~/.ssh/known_hosts
  script:
    - ssh ubuntu@$SERVER_IP "
        set -e &&
        sudo apt-get update &&
        sudo apt-get install -y docker.io &&
```

```
        sudo curl -L
\"https://github.com/docker/compose/releases/latest/download/docker-compose-
\$(uname -s)-\$(uname -m)\" -o /usr/local/bin/docker-compose &&
        sudo chmod +x /usr/local/bin/docker-compose &&
        curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.3/install.sh
| bash &&
        export NVM_DIR=\$HOME/.nvm &&
        [ -s \"\$NVM_DIR/nvm.sh\" ] && \\. \"\$NVM_DIR/nvm.sh\" &&
        nvm install 20.3.1 &&
        nvm use 20.3.1 &&
        nvm alias default 20.3.1 &&
        rm -rf /home/ubuntu/6thElement &&
        git clone https://gitlab.com/tbc6thelement/6thElement.git
/home/ubuntu/6thElement &&
        cd /home/ubuntu/6thElement &&
        sudo /usr/local/bin/docker-compose up -d &&
        cd client &&
        npm install &&
        npm run build &&
        pm2 restart all || pm2 start npm --name \"6thElement\" -- start
      "
  only:
    - main
```


docker
app container
API
MS SQL server
IOS
Fron web