TCall Infrastructure

Development Server

All the three (1. Django Backend and FastAPI backend, 2. TCall React Dashboard, and 3. TCall Marketing website) repository deployed on the same ec2 instance. (For the saving the cost)

Region: il-central-1

Frontend URL (React.js): dev.dashboard.tcall.ai

Backend URL: api.dev.tcall.ai

Note: Domain is hosted in Route53 and uses Let's Encrypt SSL.

EC2 Instance: i-04b3a276864e59b4a (FE & BE Dev-Tcall)
Public IP: 51.84.90.58
Keypair: dev-tcall.pem

RDS Details:

DB_NAME=tcall_dev_db_v1

DB_USER=postgres

DB_PASSWORD=xwT78sr7n0ll

DB_HOST=dev-tcall.czouassyu7f2.il-central-1.rds.amazonaws.com

DB_PORT=5432

S3 Bucket: tcall-dev-bucket

Note: The .env-folder is kept in private mode, while all other folders are in public mode to allow access to images in the same S3 bucket. During the CI/CD process, .env files are fetched from the S3 bucket to the Dev EC2 instance.

Note: Fronted working though PM2, And Backend using the supervisor and Gunicorn

```
ubuntu@ip-172-31-6-66:/etc/nginx/sites-available$ pm2 list
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
2	react-frontend	default	N/A	fork	49122	2D	7	online	0%	72.2mb	ubuntu	disabled

```
ubuntu@ip-172-31-6-66:/etc/nginx/sites-available$
```

Project root directory

```
Last login: Fri Dec 27 05:03:16 2024 from 152.59.192.197
ubuntu@ip-172-31-6-66:~$ ls
Tcall  aws  awscliv2.zip  cnp2098_tcall_frontend  gitlab-runner_amd64.deb
ubuntu@ip-172-31-6-66:~$
```

Frontend: /home/ubuntu/cnp2098_tcall_frontend

Backend Django: /home/ubuntu/Tcall

Backend FastAPI: /home/ubuntu/ai-dev-tcall-ai/cnp2098_tcall_backend/11labs

---

Production/Live Server

Backend Detail:

TCall Backend (Django App and Fast API)

EC2 Instance: *i-0286cd36a42e04e69* (BE-Prod-tcalll)
Public IP: 51.17.217.45
Keypair: *tcall_dev.pem*

RDS Details:

DB_NAME=tcall_dev_db_v1

DB_USER=postgres

DB_PASSWORD=Ao4OVQQXZtLRDW6jAtgG

DB_HOST=tcall-rds-db.czouassyu7f2.il-central-1.rds.amazonaws.com

DB_PORT=5432

Project root directory:

Backend Django: /home/ubuntu/Tcall

Backend FastAPI: /home/ubuntu/ai-tcall-ai/cnp2098_tcall_backend/11labs

Frontend Details:

TCall Frontend (React Dashboard)

EC2 Instance: *i-i-01bb8b1db81931976* (FE-Prod-tcalll)
Public IP: 51.84.62.215
Keypair: *tcall_dev.pem / Password: Tcall@2024#*

Project root directory:

Frontend: /home/ubuntu/cnp2098_tcall_frontend

S3 Bucket: tcall-prod-bucket

Note: The .env-folder is kept in private mode, while all other folders are in public mode to allow access to images in the same S3 bucket. During the CI/CD process, .env files are fetched from the S3 bucket to the Dev EC2 instance.

Note: Fronted working though PM2, And Backend using the supervisor and Gunicorn

Frontend CI/CD YAML File:

```
stages:
  - deploy

deploy_to_master:
  stage: deploy
  tags:
    - ubuntu
```

```
  script:
    - echo "Sourcing profile..."
    - echo "Checking for existing repository in /home/gitlab-runner..."
    # Check if the repo exists and remove it before cloning
    - whoami
    - sudo -i
    #- if [ -d "/home/ubuntu/cnp2098_tcall_frontend" ]; then rm -rf /home/ubuntu/cnp2098_tcall_frontend; fi
    # Clone the repository into /home/gitlab-runner
    #- cd /home/ubuntu
    #- git clone http://smruti:tNbx8QsT_V7MEin_2fr6@ec2-52-7-13-83.compute-1.amazonaws.com/deepkumarsaha/cnp2098_tcall_frontend.git /home/ubuntu/cnp2098_tcall_frontend
    # Navigate to the cloned directory
    - cd /home/ubuntu/cnp2098_tcall_frontend
    - echo "Checking out repository..."
    - git checkout .
    - git checkout master
    - git pull origin master
    - git branch
    - git status
    - pwd
    #- echo "Copying .env from S3..."
    - sudo aws s3 cp s3://tcall-prod-bucket/.env-folder/frontend-env/.env /home/ubuntu/cnp2098_tcall_frontend/.env
    #- echo "Setting ownership to gitlab-runner for project files (excluding node_modules)..."
    # Change ownership of the project files, excluding node_modules
    #- chown -R ubuntu:ubuntu /home/gitlab-runner/cnp2098_tcall_frontend
    #- chmod -R 777 /home/gitlab-runner/cnp2098_tcall_frontend
```

```
    - echo "Installing dependencies..."
    # Install dependencies as gitlab-runner (avoiding permission issues with node_modules)
    - cd /home/gitlab-runner/cnp2098_tcall_frontend && npm install
    - echo "Building project..."
    # Run the build step
    - cd /home/gitlab-runner/cnp2098_tcall_frontend && npm run build && sleep 30
    - echo "Starting the application with PM2..."
    - pm2 delete t-call-frontend
    - pm2 list
    # Start the application using PM2
    - cd /home/gitlab-runner/cnp2098_tcall_frontend && pm2 start npm --name "t-call-frontend" -- run start
  only:
    - master

deploy_to_server:
  stage: deploy
  tags:
    - Development
  script:
    - echo "Sourcing profile..."
    - echo "Checking for existing repository in /home/gitlab-runner..."
    # Check if the repo exists and remove it before cloning
    - whoami
    #- sudo -i
    #- if [ -d "/home/ubuntu/cnp2098_tcall_frontend" ]; then rm -rf /home/ubuntu/cnp2098_tcall_frontend; fi
    # Clone the repository into /home/gitlab-runner
```

```
    #- cd /home/ubuntu

    #- git clone http://smruti:tNbx8QsT_V7MEin_2fr6@ec2-52-7-13-83.compute-1.amazonaws.com/deepkumarsaha/cnp2098_tcall_frontend.git /home/ubuntu/cnp2098_tcall_frontend

    # Navigate to the cloned directory

    - cd /home/ubuntu/cnp2098_tcall_frontend

    - echo "Checking out repository..."

    - git checkout .

    - git checkout dev-environment

    - git pull

    - git branch

    - git status

    #- echo "Copying .env from S3..."

    - aws s3 cp s3://tcall-dev-bucket/.env-folder/frontend-env/.env /home/ubuntu/cnp2098_tcall_frontend/.env

    #- echo "Setting ownership to gitlab-runner for project files (excluding node_modules)..."

    # Change ownership of the project files, excluding node_modules

    #- chown -R ubuntu:ubuntu /home/gitlab-runner/cnp2098_tcall_frontend

    #- chmod -R 777 /home/gitlab-runner/cnp2098_tcall_frontend

    - echo "Installing dependencies..."

    # Install dependencies as gitlab-runner (avoiding permission issues with node_modules)

    - cd /home/ubuntu/cnp2098_tcall_frontend && npm install

    - echo "Building project..."

    # Run the build step

    - cd /home/ubuntu/cnp2098_tcall_frontend && npm run build && sleep 30

    - echo "Starting the application with PM2..."

    #- sudo chown -R ubuntu:ubuntu /home/ubuntu/.config

    #- sudo chmod -R 755 /home/ubuntu/.config
```

```
    # Delete or restart the application with PM2
    - pm2 delete t-call-frontend-dev || echo "Process not found"
    #- pm2 delete t-call-frontend-dev
    - pm2 list
    # Start the application using PM2
    - cd /home/ubuntu/cnp2098_tcall_frontend && pm2 start npm --name "t-call-frontend-dev" -- run start
  only:
    - dev-environment
```

Backend CI/CD File:

```
stages:
  - deploy

deploy_to_server:
  stage: deploy
  tags:
    - dev-backend-runner
  script:
    - echo "Sourcing profile..."
    - echo "Updating the codebase..."

    # Add SSH private key to SSH agent
    - mkdir -p ~/.ssh
    - echo "$SSH_DEV_BACKEND" > ~/.ssh/id_rsa
```

```
    - chmod 600 ~/.ssh/id_rsa

    # Optional: Add server to known hosts to avoid SSH authenticity prompt
    - ssh-keyscan -H 51.84.90.58 >> ~/.ssh/known_hosts

    # Sync the code from the repository to the server (excluding unwanted files)
    - rsync -avz --no-t --exclude '.git' --exclude 'command.sh' --exclude '.env' --exclude '.gitignore' -e "ssh -i ~/.ssh/id_rsa" $CI_PROJECT_DIR/ ubuntu@51.84.90.58:/home/ubuntu/Tcall/
    - aws s3 cp s3://tcall-prod-bucket/.env-folder/Backend-env/.env /home/ubuntu/Tcall/.env

    # Install dependencies (virtualenv, if necessary)
    - |
      if [ ! -d "/home/ubuntu/Tcall/tenv" ]; then
        python3 -m venv /home/ubuntu/Tcall/tenv
        source /home/ubuntu/Tcall/tenv/bin/activate
        pip install -r /home/ubuntu/Tcall/ubuntu_requirments.txt
      fi
      echo "Created new venv"
      source /home/ubuntu/Tcall/tenv/bin/activate

      if [ "$is_dependeny_update" == "true" ]; then
        pip install -r /home/ubuntu/Tcall/ubuntu_requirments.txt
      fi

    # Run migrations if is_migrate is set to true
    - |
```

```
    if [ "$is_migrate" == "true" ]; then
      echo "Running Django migrations..."
      cd /home/ubuntu/Tcall  # Change to the directory where your Django project is located
      source tenv/bin/activate
      python3 manage.py makemigrations  # Create new migrations if needed
      python3 manage.py migrate  # Apply migrations to the database
    else
      echo "Skipping migrations as is_migrate is not true."
    fi

  # Restart Supervisor
  - echo "Restarting Supervisor services..."
  - sudo /usr/bin/systemctl restart supervisor

 only:
  - dev-environment  # Trigger when changes are pushed to the dev-environment branch

deploy_to_prod:
 stage: deploy
 tags:
  - ubuntu
 script:
  - echo "Sourcing profile..."
  - echo "Updating the codebase..."

  # Add SSH private key to SSH agent
```

```
    - mkdir -p ~/.ssh

    - echo "$SSH_PROD_BACKEND" > ~/.ssh/id_rsa

    - chmod 600 ~/.ssh/id_rsa


    # Optional: Add server to known hosts to avoid SSH authenticity prompt

    - ssh-keyscan -H 51.17.217.45 >> ~/.ssh/known_hosts


    # Sync the code from the repository to the server (excluding unwanted files)

    - rsync -avz -v --no-t --exclude '.git' --exclude 'command.sh' --exclude '.env' --exclude '.gitignore' -e "ssh -i ~/.ssh/id_rsa" $CI_PROJECT_DIR/ ubuntu@51.17.217.45:/home/ubuntu/Tcall/ --rsync-path="sudo rsync"

    - aws s3 cp s3://tcall-dev-bucket/.env-folder/Backend-env/.env /home/ubuntu/Tcall/.env


    # Install dependencies (virtualenv, if necessary)

    - if [ ! -d "/home/ubuntu/Tcall/tenv" ]; then python3 -m venv /home/ubuntu/Tcall/tenv && source /home/ubuntu/Tcall/tenv/bin/activate && pip install -r /home/ubuntu/Tcall/ubuntu_requirments.txt; fi

    - echo "Created new venv"

    - source /home/ubuntu/Tcall/tenv/bin/activate


    - if [ "$is_dependeny_update" == "true" ]; then pip install -r /home/ubuntu/Tcall/ubuntu_requirments.txt; fi


    # Run migrations if is_migrate is set to true

    - if [ "$is_migrate" == "true" ]; then echo "Running Django migrations..."; cd /home/ubuntu/Tcall && python3 manage.py makemigrations && python3 manage.py migrate; else echo "Skipping migrations as is_migrate is not true."; fi


    # Print current branch name

    #- echo "Current Branch: ${ CI_COMMIT_REF_NAME }"
```

```
  # Restart Supervisor
  - echo "Restarting Supervisor services..."
  - sudo /usr/bin/systemctl restart supervisor


only:
  - master          # Trigger when changes are pushed to the master branch
```