

Minumum Hardware Requirements

- CPU: 2vCPU
- RAM: 4GB
- Disk: 20GB

Software Requirements

- OS: Ubuntu 20.04 or later
- Docker: Latest
- Caddy Server: Latest

File Location

Production

- Docker Compose File: `~/docker-compose.yml`
- Caddy Server Configuration: `/etc/caddy/Caddyfile`
- Frontend Files: `/var/www/site`

Testing

- Docker Compose File: `~/test/docker-compose.yml`
- Caddy Server Configuration: `/etc/caddy/Caddyfile`
- Frontend Files: `/var/www/test`

Installation Steps

Docker Installation

1. Get docker Installation script

```
wget -O get-docker.sh https://get.docker.com
chmod +x get-docker.sh
```

2. Run the script

```
./get-docker.sh
```

3. Add your user to the docker group

```
sudo usermod -aG docker $USER
```

4. Login to docker

```
docker login
```

Caddy Server Installation

1. Install Caddy

```
sudo apt install -y debian-keyring debian-archive-keyring apt-transport-https
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/gpg.key' | sudo gpg --
dearmor -o /usr/share/keyrings/caddy-stable-archive-keyring.gpg
curl -1sLf 'https://dl.cloudsmith.io/public/caddy/stable/debian.deb.txt' | sudo
tee /etc/apt/sources.list.d/caddy-stable.list
sudo apt update
sudo apt install caddy
```

Configuration Steps

Frontend Site

1. Create a directory for the frontend files

```
sudo mkdir -p /var/www/site
```

2. Update ownership to caddy user and caddy group

```
sudo chown -R caddy:caddy /var/www/site
```

3. Copy the frontend files to the directory

4. Update the Caddyfile to point to the frontend files

```
ottawatamilsangam.org {
    handle /* {
        root * /var/www/site
        file_server
        try_files {path} /index.html
        encode gzip
    }
}
```

5. Reload Caddy

```
sudo systemctl reload caddy
```

Backend

1. Create a docker-compose file

```
services:
  db:
    image:
mongo@sha256:b5725ac74a0d00d2bb48af997535f4eb4e87df61a2054306ae06f258b6c6b337
    restart: always
    volumes:
      - "/home/user/test/db/data:/data/db" # Change this to your desired location
    ports:
      - 27017:27017
```

```
healthcheck:
  test: echo 'db.runCommand("ping").ok' | mongosh localhost:27017/test --
quiet
  interval: 10s
  timeout: 3s
  retries: 3
logging:
  options:
    max-size: 5M
backend:
  image: ottawatamilsangam/backend
  restart: always
  depends_on:
    db:
      condition: service_healthy
      restart: true
  environment:
    MONGODB_URI: mongodb://db:27017/ots
    SERVER_PORT: 3000
    SECRET_STR: # JWT secret
    LOGIN_EXPIRE: 2592000
    SENDGRID_API_KEY: # Sendgrid API key
  ports:
    - 3000:3000
  logging:
    options:
      max-size: 5M
```

2. Pull the docker image

```
docker compose pull
```

3. Start the docker containers

```
docker compose up -d
```