Deployment Guide

This guide provides detailed instructions for deploying the CMU-Africa Information Assistant to production.

🎯 Target Domain

Production URL: campuslink.apps.cximmersion.com

📋 Pre-Deployment Checklist

- [ ] OpenAI API key obtained and tested
- [ ] Pinecone account created with index
- [ ] Knowledge base populated with data
- [ ] Environment variables configured
- [ ] Application tested locally
- [ ] Security review completed
- [ ] Backup strategy defined

🚀 Deployment Options

Option 1: Streamlit Cloud (Recommended for Quick Deploy)

Steps:

1. **Push to GitHub**
   ```bash
   git init
   git add .
   git commit -m "Initial commit"
   git remote add origin <your-repo-url>
   git push -u origin main
   ```

2. **Deploy on Streamlit Cloud**
   - Go to share.streamlit.io (https://share.streamlit.io)
   - Click “New app”
   - Select your repository
   - Set main file path: `src/app.py`
   - Configure secrets (see below)

3. **Configure Secrets**
   In Streamlit Cloud secrets management, add:
   ```toml
   OPENAI_API_KEY = "sk-..."
   PINECONE_API_KEY = "..."
   PINECONE_ENVIRONMENT = "us-east-1-aws"
   PINECONE_INDEX_NAME = "cmu-africa-assistant"
   ```

4. **Custom Domain**
   - Go to Settings → General
   - Add custom domain: `campuslink.apps.cximmersion.com`
   - Follow DNS configuration instructions

Option 2: Docker Deployment

Using Docker Compose:

1. **Set up environment variables**
   `bash`
   `cp .env.example .env`
   `# Edit .env with your API keys`

2. **Build and run**
   `bash`
   `docker-compose up -d`

3. **Access the application**
   `http://localhost:8501`

Using Docker only:

```
# Build
docker build -t cmu-africa-assistant .

# Run
docker run -d \
  -p 8501:8501 \
  -e OPENAI_API_KEY="your_key" \
  -e PINECONE_API_KEY="your_key" \
  -e PINECONE_ENVIRONMENT="your_env" \
  --name cmu-assistant \
  cmu-africa-assistant
```

Option 3: Traditional Server Deployment

Prerequisites:

- Ubuntu 20.04+ or similar Linux distribution
- Python 3.9+
- Nginx (for reverse proxy)
- Supervisor or systemd (for process management)

Steps:

1. **Set up the server**
   ```bash
   # Update system
   sudo apt update && sudo apt upgrade -y

# Install Python and dependencies
sudo apt install python3.9 python3.9-venv python3-pip nginx -y
```

1. **Deploy application**
   ```bash
   # Clone repository
   ```

cd /opt
sudo git clone cmu-africa-assistant
cd cmu-africa-assistant

# Create virtual environment
sudo python3.9 -m venv venv
sudo venv/bin/pip install -r requirements.txt

# Set up environment
sudo cp .env.example .env
sudo nano .env # Add your API keys

# Set permissions
sudo chown -R www-data:www-data /opt/cmu-africa-assistant
```

1. **Configure Supervisor**
   Create `/etc/supervisor/conf.d/cmu-assistant.conf`:
   ```ini
   [program:cmu-africa-assistant]
   command=/opt/cmu-africa-assistant/venv/bin/streamlit run src/app.py --server.port=8501
   directory=/opt/cmu-africa-assistant
   user=www-data
   autostart=true
   autorestart=true
   redirect_stderr=true
   stdout_logfile=/var/log/cmu-assistant.log
   environment=OPENAI_API_KEY="your_key",PINECONE_API_KEY="your_key"
   ```

Start the service:
```bash
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start cmu-africa-assistant
```

1. **Configure Nginx**
   Create `/etc/nginx/sites-available/cmu-assistant`:
   ```nginx
   server {
   listen 80;
   server_name campuslink.apps.cximmersion.com;

   location / {
   proxy_pass http://localhost:8501;
   proxy_http_version 1.1;
   proxy_set_header Upgrade $http_upgrade;
   proxy_set_header Connection “upgrade”;
   proxy_set_header Host $host;
   proxy_cache_bypass $http_upgrade;
   }
   }
   ```
   ```

Enable site:

```
bash
  sudo ln -s /etc/nginx/sites-available/cmu-assistant /etc/nginx/sites-enabled/
  sudo nginx -t
  sudo systemctl reload nginx
```

1. **Set up SSL with Let's Encrypt**

   ```
   bash
     sudo apt install certbot python3-certbot-nginx -y
     sudo certbot --nginx -d campuslink.apps.cximmersion.com
   ```

🔒 Security Configuration

1. Environment Variables

Never hardcode secrets. Use environment variables:

```
export OPENAI_API_KEY="sk-..."
export PINECONE_API_KEY="..."
```

2. Firewall Configuration

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 22/tcp
sudo ufw enable
```

3. Rate Limiting

Add to Nginx configuration:

```
limit_req_zone $binary_remote_addr zone=one:10m rate=10r/s;

server {
    location / {
        limit_req zone=one burst=20;
        # ... rest of config
    }
}
```

4. Authentication (Optional)

For admin panel, add basic auth:

```
location /Admin {
    auth_basic "Admin Area";
    auth_basic_user_file /etc/nginx/.htpasswd;
    # ... proxy config
}
```

📊 Monitoring and Logging

Application Logs

Logs are stored in `logs/` directory:
- Application logs: `logs/app.log`
- Error logs: `logs/error.log`

System Monitoring

1. **Set up monitoring** with tools like:
   - Prometheus + Grafana
   - New Relic
   - DataDog

2. **Key metrics to monitor**:
   - Response time
   - Error rate
   - API usage (OpenAI, Pinecone)
   - Memory usage
   - CPU usage

Log Rotation

Configure logrotate:

```
sudo nano /etc/logrotate.d/cmu-assistant
```

```
/opt/cmu-africa-assistant/logs/*.log {
    daily
    rotate 14
    compress
    delaycompress
    notifempty
    create 0640 www-data www-data
}
```

🔄 Updates and Maintenance

Updating the Application

```
cd /opt/cmu-africa-assistant
sudo -u www-data git pull
sudo -u www-data venv/bin/pip install -r requirements.txt
sudo supervisorctl restart cmu-africa-assistant
```

Database Backups

Regular backups of chat history and feedback:

```
#!/bin/bash
# backup.sh
BACKUP_DIR="/backups/cmu-assistant"
DATE=$(date +%Y%m%d_%H%M%S)

mkdir -p $BACKUP_DIR
cp -r /opt/cmu-africa-assistant/data $BACKUP_DIR/data_$DATE
```

Add to crontab:

```
0 2 * * * /opt/cmu-africa-assistant/backup.sh
```

🧪 Testing the Deployment

Health Checks

1. **Application health**:
   `bash`
   `curl http://localhost:8501/_stcore/health`

2. **API connectivity**:
   - Test OpenAI: Check a query in the chat
   - Test Pinecone: View index stats in Admin Panel

3. **SSL certificate**:
   `bash`
   `curl -I https://campuslink.apps.cximmersion.com`

Load Testing

Use tools like Apache Bench:

```
ab -n 1000 -c 10 http://campuslink.apps.cximmersion.com/
```

🆘 Troubleshooting

Common Issues

1. **Application won't start**
   `bash`
   `sudo supervisorctl status cmu-africa-assistant`
   `sudo tail -f /var/log/cmu-assistant.log`

2. **502 Bad Gateway**
   - Check if Streamlit is running
   - Verify port 8501 is accessible
   - Check Nginx error logs

3. **Slow responses**
   - Monitor API rate limits
   - Check server resources
   - Review Pinecone index performance

Emergency Rollback

```
cd /opt/cmu-africa-assistant
sudo -u www-data git reset --hard HEAD~1
sudo supervisorctl restart cmu-africa-assistant
```

📱 Post-Deployment

1. Populate Knowledge Base

```
cd /opt/cmu-africa-assistant
sudo -u www-data venv/bin/python populate_knowledge_base.py
```

2. Test All Features

- [ ] Chat functionality
- [ ] Document search
- [ ] Admin panel
- [ ] Feedback system
- [ ] Multi-language support

3. Monitor for 24 Hours

Watch for:
- Error rates
- Response times
- API costs
- User feedback

📞 Support

For deployment issues:
- Documentation: See README.md
- Logs: Check `/var/log/` and `logs/`
- Status: `sudo supervisorctl status`

Deployment Domain: campuslink.apps.cximmersion.com
Last Updated: October 2024