Deployment Guide for ResearchLens

This guide provides instructions for deploying the ResearchLens application on an external Virtual Machine (VM).

Prerequisites

- Ubuntu 20.04 LTS or later
- Python 3.8 or higher
- PostgreSQL 12 or higher
- 4GB RAM minimum (8GB recommended)
- 20GB disk space minimum

VM Setup

1. Initial Server Setup

```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Install required system packages
sudo apt install -y python3 python3-pip python3-venv git nginx supervisor postgresql po
stgresql-contrib

# Create application user
sudo useradd -m -s /bin/bash researchlens
sudo usermod -aG sudo researchlens
```

2. Database Setup

```
# Switch to postgres user
sudo -u postgres psql

# Create database and user
CREATE DATABASE researchlens;
CREATE USER researchlens WITH PASSWORD 'your_secure_password';
GRANT ALL PRIVILEGES ON DATABASE researchlens TO researchlens;
\q

# Configure PostgreSQL
sudo nano /etc/postgresql/12/main/postgresql.conf
# Uncomment and set: listen_addresses = 'localhost'

sudo nano /etc/postgresql/12/main/pg_hba.conf
# Add line: local   researchlens    researchlens                    md5

# Restart PostgreSQL
sudo systemctl restart postgresql
```

3. Application Deployment

```
# Switch to application user
sudo -u researchlens -i

# Clone the application
git clone <repository-url> /home/researchlens/research_summary_app
cd /home/researchlens/research_summary_app

# Create virtual environment
python3 -m venv venv
source venv/bin/activate

# Install dependencies
pip install -r requirements.txt

# Create environment file
cat > .env << EOF
DATABASE_URL=postgresql://researchlens:your_secure_password@localhost/researchlens
ABACUSAI_API_KEY=your_api_key_here
EOF

# Set permissions
chmod 600 .env

# Initialize database
python setup.py
```

4. Nginx Configuration

```
# Create Nginx configuration
sudo nano /etc/nginx/sites-available/researchlens

# Add configuration:
server {
    listen 80;
    server_name your_domain.com;  # Replace with your domain

    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_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
        proxy_cache_bypass $http_upgrade;
        proxy_read_timeout 86400;
    }
}

# Enable site
sudo ln -s /etc/nginx/sites-available/researchlens /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl reload nginx
```

5. Supervisor Configuration

```
# Create supervisor configuration
sudo nano /etc/supervisor/conf.d/researchlens.conf

# Add configuration:
[program:researchlens]
command=/home/researchlens/research_summary_app/venv/bin/streamlit run app.py --serv-
er.port=8501 --server.address=localhost
directory=/home/researchlens/research_summary_app
user=researchlens
autostart=true
autorestart=true
redirect_stderr=true
stdout_logfile=/var/log/researchlens.log
environment=PATH="/home/researchlens/research_summary_app/venv/bin"

# Update supervisor
sudo supervisorctl reread
sudo supervisorctl update
sudo supervisorctl start researchlens
```

6. SSL Certificate (Optional but Recommended)

```
# Install Certbot
sudo apt install certbot python3-certbot-nginx

# Get SSL certificate
sudo certbot --nginx -d your_domain.com

# Auto-renewal
sudo crontab -e
# Add: 0 12 * * * /usr/bin/certbot renew --quiet
```

Monitoring and Maintenance

1. Application Monitoring

```
# Check application status
sudo supervisorctl status researchlens

# View logs
sudo tail -f /var/log/researchlens.log

# Restart application
sudo supervisorctl restart researchlens
```

2. Database Maintenance

```
# Backup database
sudo -u postgres pg_dump researchlens > backup_$(date +%Y%m%d).sql

# Restore database
sudo -u postgres psql researchlens < backup_20240101.sql

# Database maintenance
sudo -u postgres psql researchlens -c "VACUUM ANALYZE;"
```

3. System Updates

```
# Update system packages
sudo apt update && sudo apt upgrade -y

# Update Python packages
cd /home/researchlens/research_summary_app
source venv/bin/activate
pip install --upgrade -r requirements.txt

# Restart services
sudo supervisorctl restart researchlens
```

Security Considerations

1. Firewall Configuration

```
# Install UFW
sudo apt install ufw

# Configure firewall
sudo ufw default deny incoming
sudo ufw default allow outgoing
sudo ufw allow ssh
sudo ufw allow 'Nginx Full'
sudo ufw enable
```

2. Application Security

```
# Set proper file permissions
sudo chown -R researchlens:researchlens /home/researchlens/research_summary_app
sudo chmod -R 755 /home/researchlens/research_summary_app
sudo chmod 600 /home/researchlens/research_summary_app/.env

# Configure secure headers in Nginx
sudo nano /etc/nginx/sites-available/researchlens
# Add to server block:
add_header X-Frame-Options DENY;
add_header X-Content-Type-Options nosniff;
add_header X-XSS-Protection "1; mode=block";
add_header Strict-Transport-Security "max-age=31536000; includeSubDomains";
```

3. Database Security

```
# Secure PostgreSQL
sudo nano /etc/postgresql/12/main/postgresql.conf
# Set: ssl = on
# Set: password_encryption = scram-sha-256

# Restart PostgreSQL
sudo systemctl restart postgresql
```

Backup and Recovery

1. Automated Backups

```
# Create backup script
sudo nano /home/researchlens/backup.sh

#!/bin/bash
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/home/researchlens/backups"
mkdir -p $BACKUP_DIR

# Database backup
sudo -u postgres pg_dump researchlens > $BACKUP_DIR/db_backup_$DATE.sql

# Application backup
tar -czf $BACKUP_DIR/app_backup_$DATE.tar.gz /home/researchlens/research_summary_app

# Clean old backups (keep last 30 days)
find $BACKUP_DIR -name "*.sql" -mtime +30 -delete
find $BACKUP_DIR -name "*.tar.gz" -mtime +30 -delete

# Make executable
sudo chmod +x /home/researchlens/backup.sh

# Add to crontab
sudo crontab -e
# Add: 0 2 * * * /home/researchlens/backup.sh
```

2. Recovery Procedures

```
# Restore database
sudo -u postgres psql researchlens < /home/researchlens/backups/
db_backup_YYYYMMDD_HHMMSS.sql

# Restore application
tar -xzf /home/researchlens/backups/app_backup_YYYYMMDD_HHMMSS.tar.gz -C /

# Restart services
sudo supervisorctl restart researchlens
```

Performance Optimization

1. Database Optimization

```
# Configure PostgreSQL for performance
sudo nano /etc/postgresql/12/main/postgresql.conf

# Recommended settings:
shared_buffers = 256MB
effective_cache_size = 1GB
work_mem = 4MB
maintenance_work_mem = 64MB
max_connections = 100
```

2. Application Optimization

```
# Configure Streamlit for production
nano /home/researchlens/research_summary_app/.streamlit/config.toml

[server]
maxUploadSize = 200
maxMessageSize = 200
enableCORS = false
enableXsrfProtection = true

[browser]
gatherUsageStats = false

[theme]
primaryColor = "#2563EB"
backgroundColor = "#FFFFFF"
secondaryBackgroundColor = "#F8F9FA"
textColor = "#1F2937"
```

Troubleshooting

Common Issues

1. **Application won't start**
   - Check supervisor logs: `sudo tail -f /var/log/researchlens.log`
   - Verify environment variables in `.env`
   - Check database connectivity

2. **Database connection issues**
   - Verify PostgreSQL is running: `sudo systemctl status postgresql`
   - Check database credentials
   - Review PostgreSQL logs: `sudo tail -f /var/log/postgresql/postgresql-12-main.log`

3. **Nginx errors**
   - Check Nginx configuration: `sudo nginx -t`
   - Review Nginx logs: `sudo tail -f /var/log/nginx/error.log`

4. **SSL certificate issues**
   - Check certificate status: `sudo certbot certificates`
   - Renew certificate: `sudo certbot renew`

Performance Issues

1. **Slow database queries**
   - Check database indexes
   - Run VACUUM ANALYZE
   - Monitor query performance

2. **High memory usage**
   - Monitor with `htop` or `top`
   - Adjust Streamlit configuration
   - Consider upgrading server resources

Maintenance Schedule

Daily

- Monitor application logs
- Check system resources
- Verify backup completion

Weekly

- Review security logs
- Update system packages
- Check disk space usage

Monthly

- Analyze application performance
- Review database performance
- Update application dependencies
- Security audit

This deployment guide provides a comprehensive setup for production deployment of ResearchLens on an external VM. Adjust configurations based on your specific requirements and security policies.