Ollama Tunnel Configuration via Twingate

Overview

This document explains how to configure Allice to connect to Ollama running on your Mac via Twingate tunnel.

Prerequisites

1. Twingate installed and configured on your Mac
2. Ollama running on your Mac (typically on port 11434)
3. Twingate network configured to allow connections to your Mac

Configuration Steps

1. Set Up Twingate on Your Mac

```
# Install Twingate (if not already installed)
brew install --cask twingate

# Start Twingate
open -a Twingate

# Authenticate with your Twingate network
# Follow the prompts in the Twingate app
```

2. Configure Ollama to Accept Remote Connections

By default, Ollama only listens on localhost. You need to configure it to accept connections from your Twingate network.

```
# Set Ollama to listen on all interfaces
export OLLAMA_HOST=0.0.0.0:11434

# Start Ollama
ollama serve
```

Or, create a systemd service (Linux) or launchd service (Mac) to make this permanent.

For macOS (launchd):

```
# Create launchd plist file
cat > ~/Library/LaunchAgents/com.ollama.server.plist << EOF
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/Prop-
ertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.ollama.server</string>
    <key>ProgramArguments</key>
    <array>
        <string>/usr/local/bin/ollama</string>
        <string>serve</string>
    </array>
    <key>EnvironmentVariables</key>
    <dict>
        <key>OLLAMA_HOST</key>
        <string>0.0.0.0:11434</string>
    </dict>
    <key>RunAtLoad</key>
    <true/>
    <key>KeepAlive</key>
    <true/>
</dict>
</plist>
EOF

# Load the service
launchctl load ~/Library/LaunchAgents/com.ollama.server.plist
```

3. Get Your Mac’s Twingate IP Address

```
# Find your Twingate IP
ifconfig | grep -A 1 utun

# Or check in Twingate app settings
# Typically shows as something like 100.x.x.x
```

4. Configure Allice to Use Twingate Tunnel

Update your `config.json` in the Allice project:

```
{
  "models": {
    "ollama": {
      "modelWrapper": "AModelOllama",
      "baseURL": "http://<YOUR_MAC_TWINGATE_IP>:11434/v1",
      "modelList": {
        // Your models here
      }
    }
  }
}
```

Example:

```
{
  "models": {
    "ollama": {
      "modelWrapper": "AModelOllama",
      "baseURL": "http://100.64.1.25:11434/v1",
      "modelList": {
        "llama3:2": { "formatter": "AModelOllama", "contextWindow": 8192, "systemAs-
User": false }
      }
    }
  }
}
```

5. Alternative: Using Twingate Service Account

For production deployments, you can use a Twingate Service Account:

1. **Create a Service Account in Twingate:**
   - Go to Twingate admin console
   - Navigate to Settings > Service Accounts
   - Create a new service account
   - Generate service key

2. **Install Twingate Connector on Cloud Run:**

```
# Add to your Dockerfile
RUN apt-get update && apt-get install -y curl
RUN curl -o /usr/local/bin/twingate https://binaries.twingate.com/connector/linux-
amd64/twingate
RUN chmod +x /usr/local/bin/twingate
```

1. **Set Environment Variables:**

```
# In your .env or Cloud Run environment
TWINGATE_SERVICE_KEY=<your_service_key>
TWINGATE_NETWORK=<your_network_name>
```

1. **Start Twingate in Docker:**

Create a startup script:

```
#!/bin/bash

# Start Twingate connector
twingate start --service-key=$TWINGATE_SERVICE_KEY &

# Wait for connection
sleep 5

# Start AIlice
python fastapi_app.py
```

6. Testing the Connection

```
# Test from AIlice container
curl http://<YOUR_MAC_TWINGATE_IP>:11434/api/tags

# Should return list of available models
```

Environment Variables

Add these to your `.env` file:

```
# Ollama Configuration
OLLAMA_BASE_URL=http://100.64.1.25:11434/v1
OLLAMA_TUNNEL_TYPE=twingate

# Twingate Configuration (for service accounts)
TWINGATE_SERVICE_KEY=your_service_key_here
TWINGATE_NETWORK=your_network_name
```

Troubleshooting

Connection Issues

1. **Check Twingate is running:**
   `bash`
   `# On Mac`
   `twingate status`

2. **Check Ollama is accessible:**
   `bash`
   `curl http://localhost:11434/api/tags`

3. **Check firewall settings:**
   - Ensure port 11434 is not blocked
   - Check macOS firewall settings

4. **Verify Twingate network policies:**
   - Ensure your service/user has access to the Mac resource
   - Check Twingate admin console for connection logs

Performance Optimization

1. **Use local models when possible:**
   - Keep frequently used models in cloud
   - Use tunnel only for specialized Mac-only models

2. **Monitor latency:**
   - Twingate adds ~10-50ms latency
   - Consider this for real-time applications

3. **Bandwidth considerations:**
   - Large model responses may be slower
   - Consider caching strategies

Security Best Practices

1. **Use service accounts for production:**
   - Don’t rely on user authentication
   - Rotate service keys regularly

2. **Limit network access:**
   - Configure Twingate policies to restrict access
   - Only allow necessary services

3. **Monitor connections:**
   - Review Twingate audit logs
   - Set up alerts for unusual activity

4. **Keep software updated:**
   - Update Twingate client/connector regularly
   - Update Ollama regularly

References

- Twingate Documentation (https://docs.twingate.com/)
- Ollama Documentation (https://github.com/ollama/ollama)
- Allice Configuration Guide (./README.md)