Ollama Cloudflare Tunnel — Exposing Local Inference to Cloud

Ollama Cloudflare Tunnel — Exposing Local Inference to Cloud

> **Domain note (2026-05-17):** This doc refers to `ollama.basicconsulting.no` — the legacy hostname. Current live endpoint: `ollama.alai.no`. Historical examples below retain original hostname for accuracy.

Owner: FlowForge (infra)
Implemented: 2026-04-18
Purpose: Expose Mac Studio Ollama (FORGE 10.0.0.2:11434) to Azure VM LightRAG via Cloudflare tunnel with Zero Trust IP whitelist

---

Why This Tunnel

Problem: LightRAG migrated to Azure VM to eliminate Docker Desktop single point of failure. But Ollama inference stays on Mac Studio (FORGE hardware, 40 local models including q8_0 quantizations).

Solution: Cloudflare tunnel from Mac Studio exposes `ollama.basicconsulting.no` → FORGE Ollama. Azure VM LightRAG calls this endpoint for LLM/embedding inference.

Trade-off:

- Keep inference on powerful local hardware (M2 Ultra, 192GB RAM, 76 vCPU)

- □ Avoid Azure GPU VM costs ($500-2000/month)
- □ Zero refactor of LightRAG (just swap URL)
- ⚠ Mac Studio uptime now affects cloud LightRAG availability (historically 99%+, acceptable)
- ⚠ Tunnel becomes critical path (mitigated by Cloudflare's 99.99% SLA)

Tunnel Configuration

Location: `~/.cloudflared/config.yml` (Mac Studio)

```
tunnel: 3315a609-7934-45c5-ad0c-56d86d16374d
credentials-file: /Users/makinja/.cloudflared/3315a609-7934-45c5-ad0c-56d86d16374d.json

ingress:
  # ... other services ...

  - hostname: ollama.basicconsulting.no
    service: http://10.0.0.2:11434

  - service: http_status:404
```

Key points:

- `10.0.0.2` is FORGE (dedicated Ollama host on local network)
- NOT `localhost:11434` (that's ANVIL, fewer models)
- DNS: `ollama.basicconsulting.no` CNAME → `3315a609-7934-45c5-ad0c-56d86d16374d.cfargotunnel.com`

Restart tunnel after config changes:

```
launchctl kickstart -k gui/$(id -u)/com.john.cloudflared
```

Verify tunnel is up:

```
ps aux | grep cloudflared
curl https://ollama.basicconsulting.no/api/tags
# Should return JSON list of models
```

Zero Trust Policy — IP Whitelist

Policy Name: "Ollama Azure VM Only"
Application: `ollama.basicconsulting.no`
Type: Bypass (wildcard) + IP restrictions

Why bypass instead of strict Zero Trust auth:

- Pragmatic choice for initial implementation
- Existing Cloudflare setup used bypass for other internal services
- IP whitelist provides sufficient security for internal infrastructure
- Future: consider service tokens if IP rotation becomes frequent

Whitelisted IPs:

1. **Azure VM egress:** Check current VM egress IP: `ssh alai-admin@20.240.61.67 'curl -s https://ifconfig.co'`
2. **Mac Studio (backup/testing):** 46.46.251.40 (residential ISP, may rotate — see maintenance)

How policy works:

1. Azure VM LightRAG makes HTTPS request to `ollama.basicconsulting.no`
2. Cloudflare edge checks source IP against whitelist
3. If match: forward to Mac Studio tunnel → FORGE Ollama
4. If no match: return 403 Forbidden

Test from Azure VM:

```
ssh alai-admin@20.240.61.67
curl -s https://ollama.basicconsulting.no/api/tags | jq '.models | length'
# Should return model count (e.g., 12)
```

Test from random IP (should fail):

```
# From any non-whitelisted location
curl https://ollama.basicconsulting.no/api/tags
# Expected: 403 Forbidden or similar
```

IP Whitelist Maintenance

CRITICAL: Mac Studio ISP IP (46.46.251.40) is residential and WILL rotate periodically. When it does, both SSH to Azure VM and direct testing from Mac Studio will fail for Ollama tunnel testing.

Check Current Mac Studio IP

```
curl https://ifconfig.co
# Use ifconfig.co or icanhazip.com
# DO NOT use ifconfig.me (returns CDN IP on some networks)
```

Update NSG Rule (for Azure VM to access Mac Studio)

```
NEW_IP=$(curl -s https://ifconfig.co)

# Update SSH rule
az network nsg rule update \
  -g rg-alai-lightrag \
  --nsg-name vm-alai-lightragNSG \
  -n default-allow-ssh \
  --source-address-prefixes "${NEW_IP}/32"

# Update LightRAG access rule (if used for direct access)
az network nsg rule update \
  -g rg-alai-lightrag \
  --nsg-name vm-alai-lightragNSG \
  -n allow-lightrag-macstudio \
  --source-address-prefixes "${NEW_IP}/32"
```

Update Cloudflare Zero Trust Policy

Option 1: Cloudflare Dashboard

1. Log in to Cloudflare Dashboard → Zero Trust
2. Navigate to Access → Applications → "Ollama Azure VM Only"
3. Edit policy → Update IP whitelist with new Mac Studio IP
4. Save (takes effect within 30s)

Option 2: Cloudflare API (for automation)

```
# Get account ID and policy ID first (see Cloudflare API docs)
# Then use PATCH to update policy rules
# (Exact curl command omitted — requires API token with Zero Trust write access)
```

Verification after update:

```
curl https://ollama.basicconsulting.no/api/tags
# Should work from new Mac Studio IP
```

Failure Modes + Detection

Failure 1: Tunnel Process Down

Symptom: `curl https://ollama.basicconsulting.no/api/tags` returns connection timeout or 502 Bad Gateway.

Diagnosis:

```
ps aux | grep cloudflared
# If no process, tunnel is down

tail -f ~/Library/Logs/cloudflared/cloudflared.log
# Check for errors
```

Fix:

```
launchctl kickstart -k gui/$(id -u)/com.john.cloudflared
# Wait 10-15s
curl https://ollama.basicconsulting.no/api/tags
```

Persistent failure: Check launchd plist:

```
launchctl list | grep cloudflared
# Should show com.john.cloudflared

# If missing, reload plist
launchctl load ~/Library/LaunchAgents/com.john.cloudflared.plist
```

Failure 2: Model Not on Target Backend

Symptom: LightRAG logs show "model qwen2.5-coder:32b-instruct-q8_0 not found" or similar.

Diagnosis:

```
curl -s https://ollama.basicconsulting.no/api/tags | jq '.models[].name'
# Check which models are exposed
```

Cause: Tunnel points to wrong Ollama backend (ANVIL vs FORGE).

Fix:

```
# Check config
cat ~/.cloudflared/config.yml | grep -A2 "ollama.basicconsulting.no"

# Should be:
# service: http://10.0.0.2:11434  (FORGE)

# If wrong (e.g., http://localhost:11434 = ANVIL):
# Edit config, fix service URL
# Restart tunnel
launchctl kickstart -k gui/$(id -u)/com.john.cloudflared
```

Historical incident: 2026-04-18 mid-migration — tunnel initially pointed to ANVIL (localhost:11434). LightRAG couldn't find q8_0 models (only on FORGE). Changed to `10.0.0.2:11434`, resolved immediately.

Failure 3: IP Whitelist Mismatch (403 Forbidden)

Symptom: Azure VM LightRAG logs show "403 Forbidden" or "Access denied" when calling Ollama endpoint.

Diagnosis:

```
# From Azure VM
ssh alai-admin@20.240.61.67
curl -v https://ollama.basicconsulting.no/api/tags 2>&1 | grep -E "HTTP|403"
# If 403, IP not whitelisted

# Check VM egress IP
```

```
curl -s https://ifconfig.co
```

Fix: Update Zero Trust policy (see IP Whitelist Maintenance section above).

Failure 4: Latency Spike (>500ms for /api/tags)

Symptom: Slow LightRAG responses; Ollama calls taking >1s for simple requests.

Diagnosis:

```
# From Azure VM
time curl -s https://ollama.basicconsulting.no/api/tags > /dev/null
# Should be 30-80ms typically

# From Mac Studio (local baseline)
time curl -s http://10.0.0.2:11434/api/tags > /dev/null
# Should be <10ms
```

Possible causes:

1. **Mac Studio network issue:** Check Wi-Fi/Ethernet, router, ISP
2. **Cloudflare edge routing:** Rare but possible; check Cloudflare status page
3. **FORGE overloaded:** Other processes using Ollama heavily

Fix 3 (FORGE overload):

```
curl http://10.0.0.2:11434/api/ps
# Check running models and concurrent requests
# Identify and throttle/stop competing workloads
```

Performance Characteristics

Expected latency (Azure swedencentral ↔ Mac Studio Oslo):

- `/api/tags` (simple): 30-80ms
- Single inference (short prompt, q8_0 model): 500ms-5s (mostly inference time, not network)
- Streaming inference: 30-60ms added to time-to-first-token

Bandwidth: Not a bottleneck. Ollama API uses JSON over HTTPS; typical request/response <100KB except for large context prompts.

Throughput: Tunnel supports multiple concurrent requests. Bottleneck is FORGE hardware, not tunnel.

Cloudflare Tunnel SLA: 99.99% uptime (per Cloudflare SLA for paid plans). ALAI on Free plan but historically stable.

Security Considerations

Current model: IP whitelist via Cloudflare Zero Trust bypass policy.

Threat model:

- □ Protects against random internet access to Ollama
- □ Restricts to known Azure VM egress IP + Mac Studio
- ⚠ If Azure VM compromised, attacker can access Ollama (acceptable — Ollama has no auth by default anyway)
- ⚠ If Mac Studio IP rotates and not updated, Azure VM loses Ollama access (operational issue, not security breach)

Future hardening options:

1. **Service tokens:** Replace IP whitelist with Cloudflare service token in request headers
2. **Mutual TLS:** Require client cert from Azure VM
3. **VPN:** Azure VNet peering to Mac Studio (complex, likely overkill)

Current assessment: IP whitelist sufficient for internal infrastructure. Service tokens recommended if IP rotation becomes operationally painful.

Monitoring

Health check (from Mac Studio):

```
curl https://ollama.basicconsulting.no/api/tags
# Should return model list
```

Health check (from Azure VM):

```
ssh alai-admin@20.240.61.67 'curl -s https://ollama.basicconsulting.no/api/tags | jq ".models | length"'
# Should return model count
```

Tunnel logs:

```
tail -f ~/Library/Logs/cloudflared/cloudflared.log
```

Cloudflare Analytics:

- Dashboard → Analytics → Traffic
- Filter by `ollama.basicconsulting.no`
- Check request count, response codes, latency percentiles

Recommended alert: If Azure VM LightRAG reports >5% Ollama request failures over 5min window, investigate tunnel status.

Related Runbooks

- **Azure LightRAG Migration:** azure-lightrag-migration.md — full migration context
- **LightRAG Backup:** lightrag-azure-backup.md — backup flow

---

Rollback / Emergency Cutover

If tunnel becomes persistently unstable:

Option 1: Move LightRAG back to Mac Studio (see azure-lightrag-migration.md rollback procedure).

Option 2: Deploy Ollama to Azure (longer-term, requires GPU VM or accept slower inference on CPU):

1. Provision Azure VM with GPU (e.g., Standard_NC4as_T4_v3, ~$500/month)
2. Install Ollama on Azure VM
3. Pull required models (qwen2.5-coder:32b-instruct-q8_0, bge-m3:latest)
4. Update LightRAG `.env`: `LLM_BINDING_HOST=http://localhost:11434`
5. Test inference latency (will be slower than FORGE M2 Ultra)

Option 3: Use Ollama Cloud / OpenAI API (cost implications, loses on-prem privacy):

- Update LightRAG to use OpenAI-compatible API
- Cost: ~$0.50-2.00 per 1M tokens (vs free on-prem)
- Latency: likely faster than current tunnel setup
- Privacy: data leaves infrastructure (requires legal review)

Recommendation: Keep current tunnel setup unless persistent failures. FORGE uptime historically excellent.

Document Owner: Skillforge
Last Updated: 2026-04-18
Validated By: Kelsey Hightower (FlowForge), Parisa Tabriz (Securion — security review)

Revision #5
Created 2026-04-18 23:16:21 UTC by John
Updated 2026-06-21 20:03:09 UTC by John