OpenRefine ↔ Wikibase Cloud Integration

This document describes **the full, working setup** for integrating **OpenRefine** with a **Wikibase Cloud** tenant, including:

- Property autocompletion in the schema editor
- Reconciliation against your own Wikibase
- Handling CORS and bot protection (Anubis)
- Docker-based reconciliation service with Redis
- TLS/SSL fixes for Alpine containers
- Final manifests and configs

This setup was validated end-to-end on **macOS**.

---

1. System Overview

Components

Component	Purpose	Address
OpenRefine	Data cleaning + schema + upload	http://127.0.0.1:3333
Wikibase Cloud	Knowledge base backend	https://wbortest.wikibase.cloud
Caddy	Local reverse proxy (CORS + Anubis bypass)	http://127.0.0.1:9999
Reconciliation Service	String → Q-ID matching	http://127.0.0.1:8000
Redis	Cache for reconciliation	Docker internal

---

2. Why Extra Infrastructure Is Needed

2.1 Property suggestions fail by default

- OpenRefine runs in the **browser** (localhost:3333)
- Wikibase Cloud runs on another domain
- Browser blocks cross-origin requests (CORS)

2.2 Wikibase Cloud uses Anubis bot protection

- Browser-like requests may be challenged
- OpenRefine cannot solve JS challenges

2.3 Reconciliation is NOT built into Wikibase

- Requires a separate **Reconciliation API** service
- Usually run via Docker

---

3. Caddy: Fixing CORS + Anubis

3.1 Install Caddy (macOS)

```
brew install caddy
```

3.2 Caddyfile

Create `~/Desktop/Caddyfile`:

```
:9999

handle /w/api.php* {
  reverse_proxy https://wbortest.wikibase.cloud {
    header_up Host wbortest.wikibase.cloud
    # Avoid Anubis browser challenge
    header_up User-Agent "OpenRefine-Caddy-Proxy/1.0"

    # CORS headers (port ≠ origin)
    header_down Access-Control-Allow-Origin *
    header_down Access-Control-Allow-Methods "GET, POST, OPTIONS"
    header_down Access-Control-Allow-Headers *
  }
}
```

3.3 Run Caddy

```
caddy run --config ~/Desktop/Caddyfile
```

3.4 Verify Caddy

```
lsof -i :9999
curl -s "http://127.0.0.1:9999/w/api.php?
action=wbsearchentities&search=instance&language=en&type=property&format=json"
```

If JSON is returned → Caddy works.

4. Reconciliation Service (Docker + Redis)

4.1 Clone the service

```
git clone https://github.com/KBNLresearch/openrefine-wikibase.git
cd openrefine-wikibase
```

4.2 docker-compose.yml

```
services:
  redis:
    image: redis:7-alpine
    restart: unless-stopped

  reconcile:
    build: .
    ports:
      - "8000:8000"
    environment:
      - REDIS_URI=redis://redis:6379/0
    depends_on:
      - redis
    restart: unless-stopped
```

4.3 Dockerfile (CRITICAL FIX)

```
FROM python:3.7-alpine

WORKDIR /openrefine-wikibase

# Fix HTTPS certificate validation
RUN apk add --no-cache ca-certificates && update-ca-certificates

RUN apk add --no-cache gcc musl-dev linux-headers libffi-dev
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

ADD . /openrefine-wikibase

EXPOSE 8000
CMD [ "python", "app.py" ]
```

4.4 config.py (Redis fix)

```
import os
redis_uri = os.getenv("REDIS_URI", "redis://redis:6379/0")
```

4.5 Start the service

```
docker compose down -v
docker compose up --build --force-recreate
```

4.6 Verify reconciliation

```
curl -s http://127.0.0.1:8000/en/api
curl -s "http://127.0.0.1:8000/en/api?query=instance"
```

No Redis or SSL errors → service is healthy.

---

5. OpenRefine Wikibase Manifest (FINAL)

```
{
  "version": "1.0",

  "mediawiki": {
    "name": "wbortest",
    "root": "https://wbortest.wikibase.cloud/wiki/",
    "main_page": "https://wbortest.wikibase.cloud/wiki/Main_Page",
    "api": "http://127.0.0.1:9999/w/api.php"
  },

  "wikibase": {
    "site_iri": "https://wbortest.wikibase.cloud/entity/",
    "maxlag": 5,
    "sparql": {
      "endpoint": "https://wbortest.wikibase.cloud/query/sparql"
    },
    "properties": {
      "instance_of": "P16",
      "subclass_of": "P18"
    }
  },

  "oauth": {
```

```
    "registration_page": "https://wbortest.wikibase.cloud/wiki/
Special:OAuthConsumerRegistration/propose"
  },

  "reconciliation": {
    "endpoint": "http://127.0.0.1:8000/${lang}/api"
  },

  "entity_types": {
    "item": {
      "name": "Item",
      "reconciliation_endpoint": "http://127.0.0.1:8000/${lang}/api"
    },
    "property": {
      "name": "Property",
      "reconciliation_endpoint": "http://127.0.0.1:8000/${lang}/api"
    }
  }
}
```

After adding the instance: - Reload OpenRefine - Property suggestions should appear - Reconciliation should work

---

6. Debugging Checklist

Property suggestions empty

- Check DevTools → Network → request goes to `127.0.0.1:9999`
- Confirm CORS headers present
- Ensure Anubis is bypassed (User-Agent override)

Reconciliation errors

- Redis running (`docker compose ps`)
- `curl http://127.0.0.1:8000/en/api`
- No `localhost:6379` in config

SSL errors

- `ca-certificates` installed in Docker image
- No `www.` in Wikibase URLs

---

7. Key Lessons

- OpenRefine runs in the browser → CORS matters
- Wikibase Cloud uses bot protection → User-Agent matters
- Reconciliation is a separate service
- Docker networking ≠ localhost
- Alpine images need CA certificates for HTTPS

This setup is production-grade and reusable for other Wikibase tenants.