Smart Contract Audit Bot - Testing Instructions

This document provides step-by-step instructions for testing the Pinecone connection and basic ingestion functionality after the configuration fixes.

Overview

The configuration issues have been resolved:
1. `.env` file formatting fixed (removed quotes and comments)
2. `config.py` now properly cleans environment variable values
3. Pinecone ServerlessSpec receives clean values

Quick Start Testing

Step 1: Set Up API Keys

1. **Copy the environment template:**
   `bash`
   `   cp .env.template .env`

2. **Edit `.env` with your real API keys:**
   `bash`
   `   nano .env  # or use your preferred editor`

3. **Replace the placeholder values:**

```env
# Replace these with your actual keys:
PINECONE_API_KEY=your_actual_pinecone_api_key
OPENAI_API_KEY=your_actual_openai_api_key

# These can stay as-is for testing:
PINECONE_ENVIRONMENT=us-east-1-aws
PINECONE_CLOUD=aws
PINECONE_REGION=us-east-1
INDEX_NAME=smart-contract-audit-test
DIMENSION=1536
```

Step 2: Run the Test Suite

Option A: Use the automated test script (Recommended)

```
bash scripts/run_tests.sh
```

Option B: Run tests manually

```
# Install dependencies
pip install pytest numpy
pip install -r requirements.txt

# Run the test suite
python tests/test_pinecone_connection.py
```

What the Tests Validate

1. Configuration Loading Tests

- Environment variables load correctly
- Value cleaning works (removes quotes, comments, whitespace)
- Required fields validation
- Proper data types (dimension as integer)

2. Pinecone Connection Tests

- Pinecone initialization with ServerlessSpec
- Index creation/connection
- Basic index statistics retrieval

3. Basic Functionality Tests

- Document ingestion (upsert operations)
- Vector search/query operations
- Document retrieval (fetch operations)
- Cleanup operations

4. Mock Tests (No API Keys Required)

- Configuration validation with mock data
- Pinecone operations simulation
- Error handling verification

Expected Test Output

With Real API Keys:

```
==========================================================
SMART CONTRACT AUDIT BOT - PINECONE FUNCTIONALITY TESTS
==========================================================
✅ Real API keys detected - running full integration tests

✅ Configuration loaded successfully
  - Environment: us-east-1-aws
  - Cloud: aws
  - Region: us-east-1
  - Index: smart-contract-audit-test
  - Dimension: 1536

✅ Environment value cleaning works correctly
✅ Pinecone initialization successful
  - Index stats: {'dimension': 1536, 'total_vector_count': 0}
✅ Test vectors upserted successfully
✅ Query executed successfully
  - Found 2 matches
✅ Fetch executed successfully
✅ Test vectors cleaned up
✅ Mock Pinecone operations work correctly

All tests passed!
```

With Template/Mock Keys:

```
==========================================================
SMART CONTRACT AUDIT BOT - PINECONE FUNCTIONALITY TESTS
==========================================================
⚠️ Template/mock API keys detected - running configuration and mock tests only
  To run full tests, update .env with real API keys

✅ Configuration loaded successfully
✅ Environment value cleaning works correctly
✅ Mock Pinecone operations work correctly

Configuration tests passed! (Integration tests skipped)
```

Troubleshooting

Common Issues and Solutions

1. Missing .env file:

```
Error: .env file not found!
```

Solution: Copy `.env.template` to `.env` and add your API keys.

2. Invalid API key format:

```
Error initializing Pinecone: Invalid API key format
```

Solution: Ensure your Pinecone API key starts with `pc-` and OpenAI key starts with `sk-`.

3. Network/connectivity issues:

```
Error: Connection timeout
```

Solution: Check your internet connection and API key permissions.

4. Index already exists:

```
Index already exists
```

Solution: This is normal - the system will use the existing index.

API Key Requirements

Pinecone API Key

- **Where to get:** https://app.pinecone.io/
- **Format:** `pc-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
- **Permissions needed:** Create/manage indexes, upsert/query vectors

OpenAI API Key

- **Where to get:** https://platform.openai.com/api-keys
- **Format:** `sk-xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx`
- **Permissions needed:** Basic API access (for future embedding generation)

Next Steps After Testing

Once tests pass successfully:

1. **Integration Testing:** Test with real smart contract data
2. **Performance Testing:** Test with larger document sets
3. **Error Handling:** Test edge cases and error scenarios
4. **Production Setup:** Configure for production environment

File Structure

```
smart-contract-audit-bot/
├── .env                           # Your API keys (create from template)
├── .env.template                  # Template for API keys
├── config.py                     # Fixed configuration module
├── tests/
│   ├── __init__.py
│   └── test_pinecone_connection.py # Comprehensive test suite
├── scripts/
│   └── run_tests.sh              # Automated test runner
└── docs/
    └── testing_instructions.md   # This file
```

Support

If you encounter issues:
1. Check the test output for specific error messages
2. Verify your API keys are correct and have proper permissions
3. Ensure all dependencies are installed
4. Check network connectivity to Pinecone and OpenAI services

The configuration fixes ensure that malformed values are no longer passed to Pinecone’s ServerlessSpec, resolving the original PineconeApiValueError issues.