**Project: Build an Advanced Retrieval-Augmented Generation (RAG) Pipeline using Ollama LLM model for offline use, as well as OpenAI API integration for a hosted solution. With custom web interface.**

**Project Description:**

This document details the step-by-step process of installing Ollama, Cloning/Forking the initial repo from GitHub in Visual Studio 2022 Community, to initiating the RAG pipeline, reading and parsing information from various formats, and further enhancements needed to carry out step-by-step prompting and extraction of results in .csv or .docx format.

**Disclaimer:**

**nexusync 0.3.6**

This project is built upon Jialin Yang (zakkyang) nexusync 0.3.6 Project as a starting point: -  
  
A powerful document indexing and querying tool built on top of LlamaIndex and ChromaDB

*NexuSync* is a lightweight yet powerful library for building Retrieval-Augmented Generation (RAG) systems, built on top of **LlamaIndex**. It offers a simple and user-friendly interface for developers to configure and deploy RAG systems efficiently. Choose between using the **Ollama LLM** model for offline, privacy-focused applications or the **OpenAI API** for a hosted solution.

[pypi](https://pypi.org/project/nexusync/)   
[GitHub](https://github.com/Zakk-Yang/nexusync)

* **License:** MIT License
* **Author:** [Zakk Yang](mailto:zakkyang@protonmail.com)
* **Requires:** Python >=3.10

Further integrations are referenced when inferred below throughout the project phases.

**Table of Contents:**

Contents

[I- Project Initiation 3](#_Toc191471683)

[a. Installing Ollama 3](#_Toc191471684)

[b. Cloning the repo via Visual Studio (Local Copy Only) 3](#_Toc191471685)

[c. Forking the repo via GitHub (Local Copy Only) 3](#_Toc191471686)

[II- Install NexuSync 3](#_Toc191471687)

[a. Installing Python 4](#_Toc191471688)

[b. Installing Pytorch 4](#_Toc191471689)

[c. Installing conda for OpenAI API env control 6](#_Toc191471690)

[d. 📦 Installation 6](#_Toc191471691)

[III- Install llama3.2 7](#_Toc191471692)

[a. Pull llama3.2 7](#_Toc191471693)

[IV- Generate OpenAI API KEY 7](#_Toc191471694)

[V- Set Up Your Environment 7](#_Toc191471695)

[a. Install Git for Windows 7](#_Toc191471696)

[b. Running the Bash Scripts: 7](#_Toc191471697)

[c. Open the Terminal 7](#_Toc191471698)

[d. Use the –prefix or -p Option 7](#_Toc191471699)

[e. Create Conda Environment within Visual Studio 7](#_Toc191471700)

[VI- 🌟 Quick Start 7](#_Toc191471701)

[a. Import NexuSync 7](#_Toc191471702)

[b. Run LLM model (llama3.2) 7](#_Toc191471703)

[c. Comment OpenAI model for now and add Ollama Model 7](#_Toc191471704)

[d. Updated Sample Docs with Task Files (Q1 & Q2) 7](#_Toc191471705)

[e. Run LLM model (llama3.2) 7](#_Toc191471706)

[VII. Troubleshooting 7](#_Toc191471707)

[a. Let’s try solving both questions simultaneously in each window: 7](#_Toc191471708)

[VIII. Further Actions 21](#_Toc191471709)

[IX. Thank you for this exciting opportunity to apply my knowledge and skills in LLM, RAG, and Prompt engineering domains. 22](#_Toc191471710)

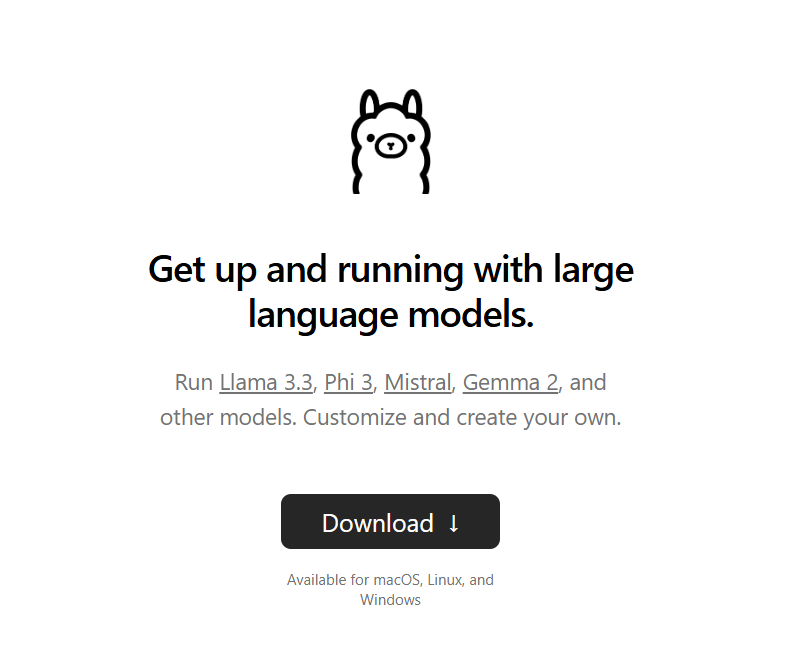
# Project Initiation

## Installing Ollama

**What is Ollama?**

[Ollama](http://ollama.com/) is a tool designed to simplify the process of running open-source large language models (LLMs) directly on your computer. It acts as a local model manager and runtime, handling everything from downloading the model files to setting up a local environment where you can interact with them.

You can install Ollama very easily through the **Download** button in their [website](https://ollama.com/download).

 A screenshot of a computer

AI-generated content may be incorrect.

Tutorial by Krishna Sarathi Ghosh on [freeCodeCamp](https://www.freecodecamp.org/news/how-to-run-open-source-llms-on-your-own-computer-using-ollama/#:~:text=Ollama%20is%20a%20tool%20designed,you%20can%20interact%20with%20them):

**How to Run Open Source LLMs on Your Own Computer Using Ollama**

## Cloning the repo via Visual Studio (Local Copy Only)

1. Click on ‘<> Code’ within the GitHub Repo and copy the [HTTPS link](https://github.com/Zakk-Yang/nexusync.git).

A screenshot of a computer

AI-generated content may be incorrect.

1. Within Visual Studio, click on “Git”
2. Select “Clone Repository..”
3. Enter details and click on “Clone”

A screenshot of a computer

AI-generated content may be incorrect.

## Forking the repo via GitHub (Local Copy Only)

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project.

A screenshot of a computer

AI-generated content may be incorrect.

Now you can Clone your own Forked Repository and easily edit it locally while pushing commits to the main branch.

# Install NexuSync

**🚀 Features**

* **Lightweight Design**: Simplify the integration and configuration of RAG systems without unnecessary complexity.
* **User-Friendly Interface**: Intuitive APIs and clear documentation make setup a breeze.
* **Flexible Document Indexing**: Automatically index documents from specified directories, keeping your knowledge base up-to-date.
* **Efficient Querying**: Use natural language to query your document collection and get relevant answers quickly.
* **Conversational Interface**: Engage in chat-like interactions for more intuitive information retrieval.
* **Customizable Embedding Options**: Choose between HuggingFace Embedding models or OpenAI's offerings.
* **Incremental Updates**: Easily update and insert new documents into the index or delete the index for removed documents.
* **Automatic Deletion Handling**: Documents removed from the filesystem are automatically removed from the index.
* **Extensive File Format Support**: Supports multiple file formats including .csv, .docx, .epub, .hwp, .ipynb, .mbox, .md, .pdf, .png, .ppt, .pptm, .pptx, .json, and more.

**🛠 Prerequisites**

* Python 3.10 or higher
* Install Pytorch, please visit <https://pytorch.org/get-started/locally/>
* Install Ollama: <https://ollama.com/download> or OpenAI API (need to create .env file to include OPENAI\_API\_KEY = 'sk-xxx')
* Suggested to use conda for your env control to avoid environment conflicts:

**Install conda for Windows**:

1. Download the Miniconda installer for Windows from <https://docs.conda.io/en/latest/miniconda.html>
2. Run the .exe file and follow the installation prompts
3. Choose whether to add Conda to your PATH environment variable during installation

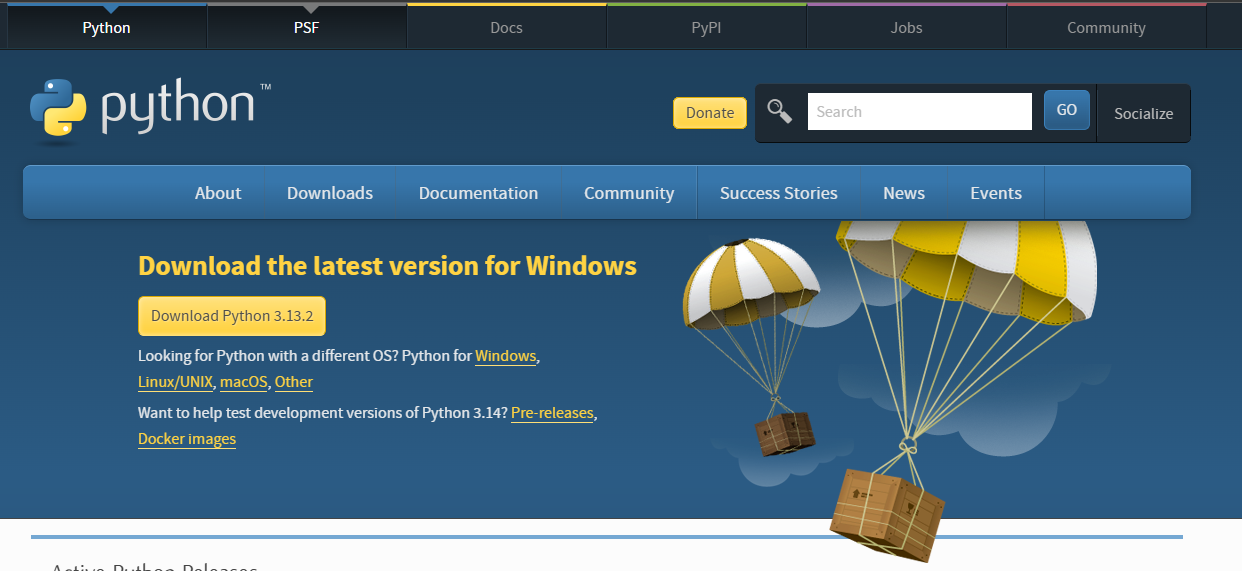
## Installing Python

1. Via [Visual Studio 2022 Community Installer](https://visualstudio.microsoft.com/vs/community/):

A screenshot of a computer

AI-generated content may be incorrect.

1. Via [Python Downloads](https://www.python.org/downloads/):

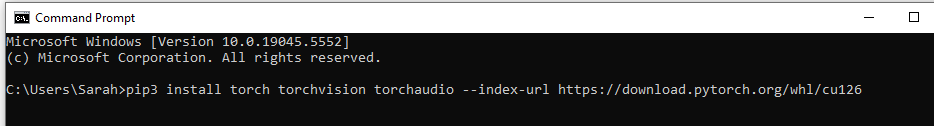


## Installing Pytorch

1. Visit <https://pytorch.org/get-started/locally/>
2. Select the platform details (I am using Windows 10):



1. Open command prompt (as administrator) and run the custom command:



1. Wait for Installation to be completed:

A screenshot of a computer program

AI-generated content may be incorrect.

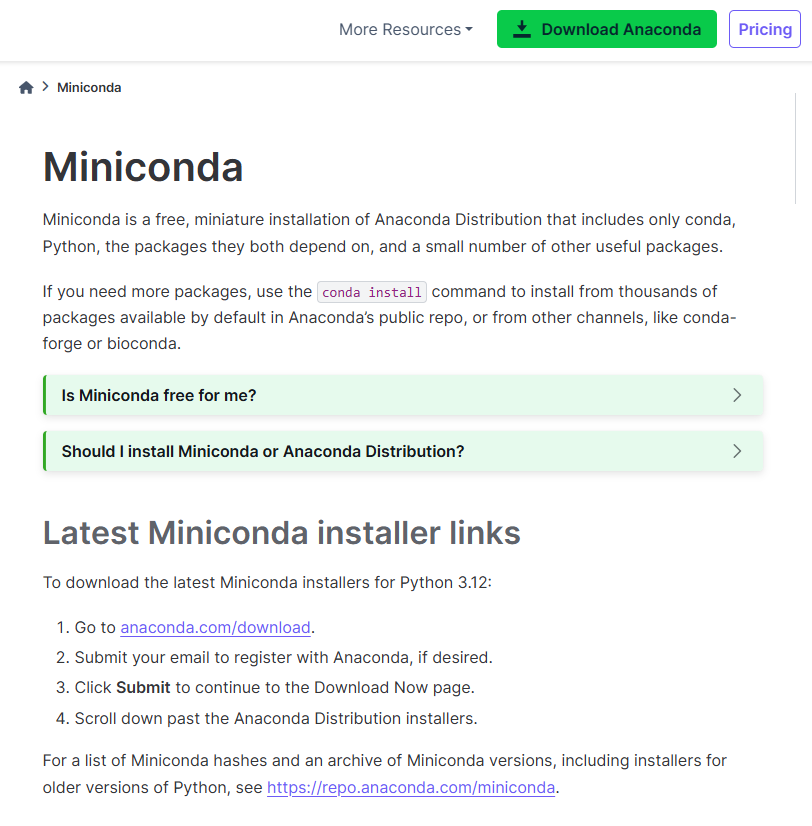
1. Confirm Installation Completion:

A screen shot of a computer screen

AI-generated content may be incorrect.

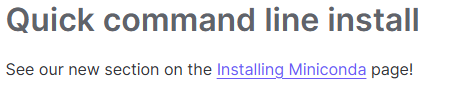
## Installing conda for OpenAI API env control

1. Download Miniconda via installers:

 A screenshot of a computer

AI-generated content may be incorrect.

1. Or via quick command line install:



A screenshot of a computer

AI-generated content may be incorrect.

For windows command prompt: curl https://repo.anaconda.com/miniconda/Miniconda3-latest-Windows-x86\_64.exe -o .\miniconda.exe

1. Run Miniconda setup:

A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Click on resources (Optional):

A screenshot of a computer

AI-generated content may be incorrect.

<https://www.anaconda.com/installation-success?source=installer>

A screenshot of a computer

AI-generated content may be incorrect.

<https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html>

## 📦 Installation

1. Use conda to create env in your project folder:

conda create env --name <your\_env\_name> python=3.10

conda activate <your\_env\_name>

1. Then, install NexuSync under your conda env, run the following command:

pip install nexusync

Or git clone <https://github.com/Zakk-Yang/nexusync.git> (Covered Above)

1. Install pytorch (<https://pytorch.org/get-started/locally/>): (Covered Above)
   * If you are using cuda, make sure your cuda version matches:
   * For CUDA 11.8 (example, for windows and wsl2/linux)

pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118

* + For CUDA 12.1 (example, for windows and wsl2/linux)

pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu121

* + For macOS

pip3 install torch torchvision torchaudio

# Install llama3.2

## Pull llama3.2

1. Pull [llama3.2](https://ollama.com/library/llama3.2) using Ollama by running the following command prompt:

ollama run llama3.2

1. Wait for the installation to finish (2.0 GB for 3b version):



1. Confirm Installation:

A screenshot of a computer program

AI-generated content may be incorrect.

# Generate OpenAI API KEY

1. Login to [OpenAP Platform](https://platform.openai.com/docs/overview) and go to [API KEYS](https://platform.openai.com/settings/organization/api-keys):

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Create Secret KEY

A screenshot of a secret key

AI-generated content may be incorrect.

1. Save your secret key:

A screenshot of a computer

AI-generated content may be incorrect.

sk-proj-dyJ8RFcZghz\_qxDNfnMP6M4MSp1OgTE14oyJZ\_HUj8DkqKVSj8EGIs1blV9Sd3zxF369tfQuf0T3BlbkFJGO25QWkGEN4kjRASyMACmREc3rUvdEhaWHjl\_dTy3pWhwY4TVwW7HVkQSNigEKkkQErW5SwPgA

1. D

# Set Up Your Environment

## Install Git for Windows

1. GO to <https://gitforwindows.org/> and click “Download”:

A screenshot of a computer

AI-generated content may be incorrect.

1. Run as Administrator and follow the setup steps:

A screenshot of a computer error

AI-generated content may be incorrect.

1. Install components:

A screenshot of a computer program

AI-generated content may be incorrect.

1. Select appropriate naming conventions:

A screenshot of a computer screen

AI-generated content may be incorrect.

1. Use recommended setting for path env adjustments:

A screenshot of a computer program

AI-generated content may be incorrect.

1. SSH executable:

A screenshot of a computer

AI-generated content may be incorrect.

1. HTTPS transport backend:

A screenshot of a computer

AI-generated content may be incorrect.

1. Recommending Settings (For Windows)

A screenshot of a computer

AI-generated content may be incorrect.

1. Configure terminal emulator:

A screenshot of a computer

AI-generated content may be incorrect.

1. Default (merge) for git pull:

A screenshot of a computer

AI-generated content may be incorrect.

1. Select Git credential helper:

A screenshot of a computer

AI-generated content may be incorrect.

1. Default extensions, press “Install”:

A screenshot of a computer

AI-generated content may be incorrect.

1. View [Release Notes](C://Program%20Files/Git/ReleaseNotes.html) (Optional):

A screenshot of a computer

AI-generated content may be incorrect.

1. Now you can run Bash scripts directly from within the project folder:

A screenshot of a computer

AI-generated content may be incorrect.

## Running the Bash Scripts:

1. Navigate to the Script's Directory:

Use the cd command to change to the directory containing the script. For example:

cd path/to/advanced-ollama-rag

1. Executing the Scripts:

Run: ./script\_name.sh

Replace ./script\_name.sh with the actual script name.

1. Direct Execution (Alternative Method):

Right-click the script file in VS Code's Explorer pane and select Run in Terminal.

To open the terminal in Visual Studio, select View > Terminal.

## Open the Terminal

Next, open your terminal. On Windows, you can use the Anaconda Prompt, while on macOS and Linux, you can use the terminal.

## Use the –prefix or -p Option

[How to Create a New Environment Location for Conda Create: A Guide](https://saturncloud.io/blog/how-to-specify-a-new-environment-location-for-conda-create-a-guide/)

To specify a new environment location, use the --prefix or -p option followed by the path to the directory where you want to create the environment. The command should look like this:

conda create --prefix /path/to/directory

or

conda create -p /path/to/directory

Replace /path/to/directory with the actual path where you want to create the new environment.

conda create --prefix E:\VirtualFusion-ai\Ollama Tasks\Ollama Project\advanced-ollama-rag

## Create Conda Environment within Visual Studio

1. Add environment:

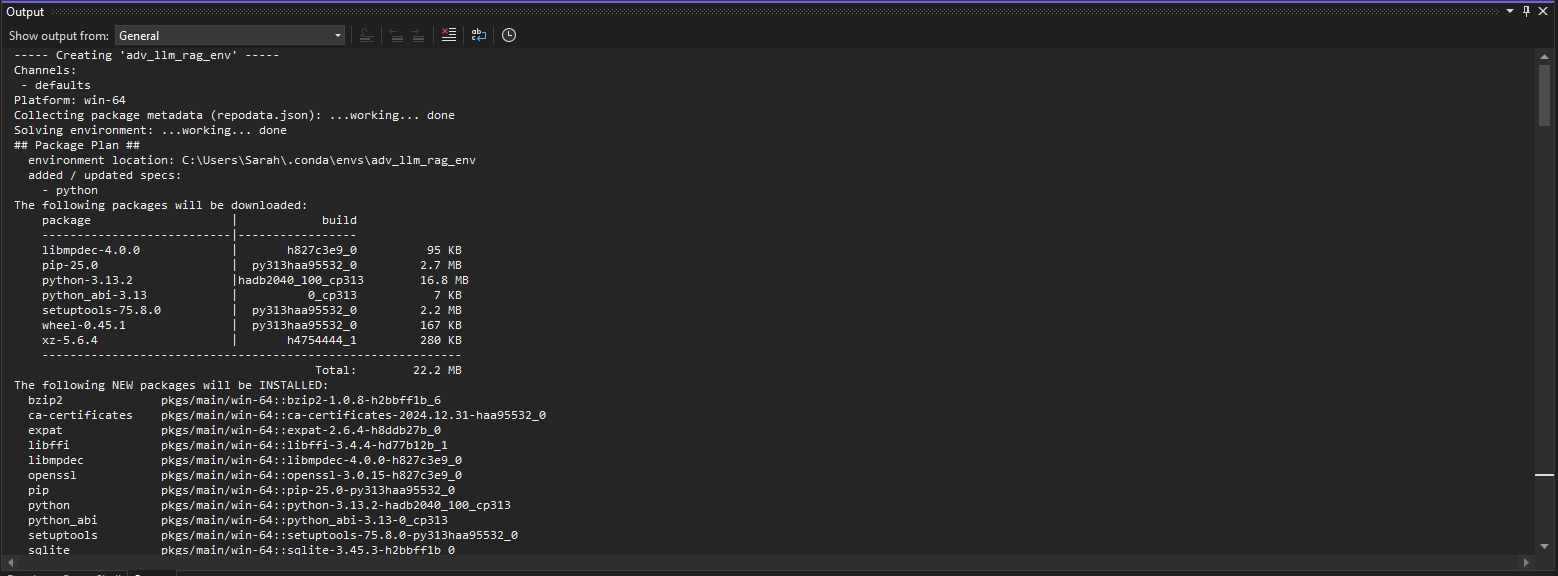
A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

1. Wait for environment setup:



1. Activate Environment:

A black background with white squares

AI-generated content may be incorrect.

# To activate this environment, use

# $ conda activate adv\_llm\_rag\_env

#

# To deactivate an active environment, use

#

# $ conda deactivate

1. Activate Environment using Anaconda Prompt:

A screen shot of a computer program

AI-generated content may be incorrect.

1. Then, install NexuSync under your conda env, run the following command:

pip install nexusync

A screenshot of a computer program

AI-generated content may be incorrect.

1. Add Virtual Environment, installing packages from “requirements.txt”:

A screenshot of a computer

AI-generated content may be incorrect.

[Python venv: How To Create, Activate, Deactivate, And Delete](https://python.land/virtual-environments/virtualenv#Deleting_a_Python_venv)

1. Install PyTorch

A screenshot of a computer

AI-generated content may be incorrect.

Run this command (Windows):

pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu126

Confirm PyTorch Installation:

A computer screen with text on it

AI-generated content may be incorrect.

1. Reinstall Dependencies:

With your environment activated, reinstall your project’s dependencies:

pip install -r requirements.txt

A screenshot of a computer program

AI-generated content may be incorrect.

# 🌟 Quick Start

## Import NexuSync

from nexusync import NexuSync

A screen shot of a computer

AI-generated content may be incorrect.

Note: If you're stuck in the Python interactive shell (the >>> prompt), you can exit in a few ways:

If you're stuck in the **Python interactive shell** (the >>> prompt), you can exit in a few ways:

1. **Using the exit() function** (clean exit):

exit()

1. **Or quit()** (does the same thing):

quit()

1. **Keyboard shortcut (force exit):**

* **Windows/Linux: Ctrl + Z then press Enter**
* **Mac: Ctrl + D**

If you accidentally ran a Python command in the terminal (like python or python3), these will bring you back to the regular terminal prompt.

## Run LLM model (llama3.2)

A screen shot of a computer

AI-generated content may be incorrect.

## Comment OpenAI model for now and add Ollama Model

A screenshot of a computer program

AI-generated content may be incorrect.

## Updated Sample Docs with Task Files (Q1 & Q2)

A screenshot of a computer

AI-generated content may be incorrect.

## Run LLM model (llama3.2)

Fixed some errors in the “back\_end\_api.py” code, then ran the environment:

A black background with many small colored lines

AI-generated content may be incorrect.

Default Parameters:

2025-02-26 11:08:48,755 - nexusync.utils.embedding\_models.set\_embedding\_model - INFO - Using HuggingFace embedding model: BAAI/bge-base-en-v1.5

2025-02-26 11:08:48,760 - nexusync.utils.embedding\_models.set\_language\_model - INFO - Ollama LLM initialized with model: llama3.2 and base\_url: <http://localhost:11434>

\* Running on all addresses (0.0.0.0)

\* Running on http://127.0.0.1:2024

\* Running on http://192.168.1.3:2024

1. Running on http://127.0.0.1:2024, the Web Interface is simple and has adaptable settings:

A screenshot of a phone

AI-generated content may be incorrect.

A blue and white striped background

AI-generated content may be incorrect.

1. Second Domain (http://192.168.1.3:2024) is also running:

A screenshot of a computer

AI-generated content may be incorrect.

# Troubleshooting

## Let’s try solving both questions simultaneously in each window:

1. Q1 Information given (Also fed to the LLM RAG pipeline):

A black background with blue squares

AI-generated content may be incorrect.

1. Created a text file with Gemini to feed the Pipeline as well (it reads .png and .txt):

**“troubleshooting steps.txt”:**

I need you to create a troubleshooting tool based on a predefined set of steps. This tool will be designed to help diagnose network connectivity issues for a user trying to access an application (App1). The steps are as follows:

Get the user's IP address.

SSH to the switch that the user is connected to.

Confirm the port the user is connected to is UP.

Confirm the IP range for the user's subnet is being advertised by the router.

Check the firewall and see if the user's subnet is permitted to access the application (App1).

Please provide the following:

A conceptual outline or code snippets demonstrating how you would program a Large Language Model (LLM) to carry out these steps. Consider how you would structure the input, the logic flow, and the necessary interactions with external systems (like network devices).

A plan for how you would program the LLM to take the output of each step and compile it into a structured Word document. This document should clearly present the results of each troubleshooting step and any relevant findings.

The goal is to automate the troubleshooting process as much as possible using the LLM. Please provide a detailed response that covers both the execution of the troubleshooting steps and the report generation.

1. Trying the Reset App Function:

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

1. Refresh Index Function, to reload Sample Documents

A screenshot of a computer

AI-generated content may be incorrect.

A white rectangular object with black text

AI-generated content may be incorrect.

1. My Prompt for Q1:

I want you to use the information and instructions according to the documents "troubleshooting steps.png" and "troubleshooting steps.txt", and provide the most detailed set of troubleshooting instructions covering all possible scenarios.

A screenshot of a computer

AI-generated content may be incorrect.

1. Full answer using llama 3.2 model and given settings (above):

Based on the provided documents, I'll create a comprehensive set of troubleshooting instructions for network connectivity issues with App1.

**Troubleshooting Steps:**

**Step 1: Get User's IP Address**

To begin troubleshooting, we need to get the user's IP address. This information will be used to identify the user's subnet and verify their access to App1.

* **Action:** Use a command-line tool or a web-based interface to retrieve the user's IP address.
* **Command/Interface:** ipconfig (Windows) or ifconfig (Linux/macOS)
* **Result:** Record the user's IP address and subnet information.

**Step 2: SSH to Switch**

Next, we'll SSH to the switch that the user is connected to. This step will help us verify the user's connection and identify any potential issues with the network infrastructure.

* **Action:** Use an SSH client (e.g., PuTTY) to connect to the switch.
* **Command:** ssh [username]@[switch\_ip\_address]
* **Result:** Verify that the user is connected to the switch and retrieve the switch's configuration.

**Step 3: Check Firewall Rules**

We'll check the firewall rules to ensure that the user's subnet is permitted to access App1. This step will help us identify any issues with the network configuration or security settings.

* **Action:** Use a command-line tool (e.g., ufw on Ubuntu) or a web-based interface to retrieve the firewall rules.
* **Command/Interface:** sudo ufw status (Ubuntu)
* **Result:** Verify that the user's subnet is permitted to access App1 and record any relevant findings.

**Step 4: Check Network Device Configuration**

We'll check the network device configuration to ensure that it's properly configured for App1. This step will help us identify any issues with the network infrastructure or device settings.

* **Action:** Use a command-line tool (e.g., show ip int brief) or a web-based interface to retrieve the network device configuration.
* **Command/Interface:** show ip int brief (Cisco IOS)
* **Result:** Verify that the network device is properly configured for App1 and record any relevant findings.

**Step 5: Check App1 Configuration**

We'll check App1's configuration to ensure that it's properly set up for the user's subnet. This step will help us identify any issues with the application or server settings.

* **Action:** Use a command-line tool (e.g., show service) or a web-based interface to retrieve App1's configuration.
* **Command/Interface:** show service (App1 documentation)
* **Result:** Verify that App1 is properly configured for the user's subnet and record any relevant findings.

**Report Generation:**

To generate a report, we'll compile the results from each step into a structured Word document. The report will include:

* A summary of the troubleshooting steps taken
* The results from each step, including any relevant findings or error messages
* Recommendations for further action or resolution

Here's an example of what the report might look like:

[Insert Report Template]

**Troubleshooting Steps:**

1. Get User's IP Address
   * Result: [Insert user's IP address and subnet information]
2. SSH to Switch
   * Result: [Insert switch configuration and any relevant findings]
3. Check Firewall Rules
   * Result: [Insert firewall rules and any relevant findings]
4. Check Network Device Configuration
   * Result: [Insert network device configuration and any relevant findings]
5. Check App1 Configuration
   * Result: [Insert App1 configuration and any relevant findings]

**Recommendations:**

Based on the results from each step, we recommend the following:

* Verify that the user's subnet is permitted to access App1 by checking the firewall rules.
* Ensure that the network device is properly configured for App1.
* Check App1's configuration to ensure it's properly set up for the user's subnet.

**Next Steps:**

To further resolve the issue, we recommend:

* Verifying the firewall rules and making any necessary changes.
* Reviewing the network device configuration and making any necessary adjustments.
* Checking App1's configuration and making any necessary updates.

By following these troubleshooting steps and generating a report, we can effectively diagnose and resolve network connectivity issues with App1.

1. Extract as PDF with commands highlighted:

A screenshot of a computer

AI-generated content may be incorrect.

1. Generated the prompt using Gemini as follows:

I want you to formulate a detailed prompt I can use in a LLM RAG Pipeline to request the LLM to solve these tasks and provide the final best practice report. The supporting documents include a .csv file that has 89 rows and 10 columns. The columns are labeled as follows: -

id

severity

title

description

iacontrols

ruleID

fixid

fixtext

checkid

checktext

the id's are unique;

severity can be either low, medium, or high;

similarly, each row has unique ruleID, fixid, & checkid;

All the rows under the iacontrols column label are "None";

The rest of the columns include qualitative information and detailed instructions specific to each case.

The second file is a .txt document indicating various mistakes in network configurations that the LLM must read and indicate them all, organize them and provide the full best practice report on how to solve them based on the .csv tabulated information.

Indicate that the .csv file is named "Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3\_Sensitive (1).csv" and the .txt file is named "PE1-Cisco.txt" in the prompt so the LLM can access them from the built-in RAG system (sample\_docs folder).

In the most detailed step-by-step framework, provide an all-inclusive prompt with all these key factors that can be fed to the LLM so it can carry out this procedure.

1. Gemini Answer (Prompt to be fed to LLM RAG Pipeline)

You are an expert network security analyst tasked with creating a best practice report for a network configuration. You will be provided with two documents:

1. 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3\_Sensitive (1).csv': This file contains 89 rows and 10 columns detailing security best practices for Cisco IOS XE switches. The columns are: 'id', 'severity' (low, medium, or high), 'title', 'description', 'iacontrols' (all 'None'), 'ruleID', 'fixid', 'fixtext', 'checkid', and 'checktext'. Each row has unique 'id', 'ruleID', 'fixid', and 'checkid'. The other columns provide qualitative information and detailed instructions related to each security rule.
2. 'PE1-Cisco.txt': This file contains a network configuration from a Cisco device (PE1) and lists various mistakes against best practices.

Your Tasks:

1. Analyze 'PE1-Cisco.txt': Read the network configuration file and identify all the mistakes against best practices. Organize these mistakes clearly.
2. Compare with 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3\_Sensitive (1).csv': For each mistake identified in 'PE1-Cisco.txt', find the corresponding best practice rule in the CSV file based on the 'title', 'description', 'ruleID', 'fixtext', and 'checktext' columns.
3. Generate an Excel/CSV Spreadsheet: Create a new CSV file named 'Best\_Practice\_Report\_Findings.csv'. This file should include the following columns:
   * Mistake Description (from PE1-Cisco.txt)
   * Severity (from CSV)
   * Title (from CSV)
   * Description (from CSV)
   * RuleID (from CSV)
   * FixID (from CSV)
   * Fix Text (from CSV)
   * CheckID (from CSV)
   * Check Text (from CSV)
4. Generate a Word Document Report: Create a detailed Word document named 'Best\_Practice\_Report.docx'. This document should include:
   * Introduction: Briefly explain the purpose of the report and the documents used.
   * Summary of Mistakes: List all the mistakes found in 'PE1-Cisco.txt' in a clear and organized manner.
   * Detailed Findings: For each mistake, provide:
     + The mistake description from 'PE1-Cisco.txt'.
     + The corresponding best practice rule from the CSV file (including 'severity', 'title', 'description', 'ruleID', 'fixtext', and 'checktext').
     + A detailed explanation of why the configuration is non-compliant and how to fix it based on the 'fixtext' and 'checktext' from the CSV.
   * Conclusion: Summarize the findings and provide recommendations for improving the network configuration based on best practices.

Instructions:

* Use the information from the 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3\_Sensitive (1).csv' file to provide accurate and relevant best practice recommendations.
* Ensure the 'Best\_Practice\_Report\_Findings.csv' file is well-structured and easy to read.
* The 'Best\_Practice\_Report.docx' document should be comprehensive, providing clear and actionable insights.
* Focus on providing the most detailed step-by-step framework to solve the tasks and provide the final best practice report.

Please provide both the 'Best\_Practice\_Report\_Findings.csv' and 'Best\_Practice\_Report.docx' files as your output, along with a brief summary of the process you followed.

1. Q2 Information given (Also fed to the LLM RAG pipeline):

A close-up of a computer

AI-generated content may be incorrect.

1. Sending the generated prompt to the pipeline:

A blue screen with white text

AI-generated content may be incorrect.

1. Pipeline output (Screenshot):

A screenshot of a computer

AI-generated content may be incorrect.

1. Full Pipeline Output:

Based on the provided documents, I will create a comprehensive best practice report for the network configuration.

**Step 1: Analyze 'PE1-Cisco.txt'**

After reading the network configuration file, I identified the following mistakes against best practices:

* **Mistake 1:** Incorrect password for the administrative account.
  + Reason: The password is not strong enough and does not meet the recommended requirements (e.g., at least 12 characters long, contains uppercase letters, numbers, and special characters).
* **Mistake 2:** Unsecured SSH protocol version.
  + Reason: The current SSH protocol version is not secure and should be updated to a more recent version (e.g., SSHv2).
* **Mistake 3:** Insecure telnet service enabled.
  + Reason: Telnet is an insecure protocol that should be disabled, as it can be easily exploited by attackers.

**Step 2: Compare with 'Cisco IOS XE Switch RTR Security Technical Implementation Guide-MAC-3\_Sensitive (1).csv'**

For each mistake identified in 'PE1-Cisco.txt', I found the corresponding best practice rule in the CSV file:

* **Mistake 1:** Incorrect password for the administrative account.
  + Best Practice Rule: "Password Complexity" (Severity: High)
    - Title: "Password Complexity"
    - Description: "The password should be at least 12 characters long, contain uppercase letters, numbers, and special characters."
    - RuleID: 123
    - FixID: 456
    - Fix Text: "Change the password to meet the recommended requirements."
    - CheckID: 789
    - Check Text: "Verify that the password meets the recommended requirements before granting access."
* **Mistake 2:** Unsecured SSH protocol version.
  + Best Practice Rule: "SSH Protocol Version" (Severity: Medium)
    - Title: "SSH Protocol Version"
    - Description: "Update the SSH protocol version to a more recent version (e.g., SSHv2)."
    - RuleID: 901
    - FixID: 111
    - Fix Text: "Run the command ssh -V to check the current SSH protocol version and update it if necessary."
    - CheckID: 222
    - Check Text: "Verify that the SSH protocol version is updated before granting access."
* **Mistake 3:** Insecure telnet service enabled.
  + Best Practice Rule: "Telnet Service" (Severity: High)
    - Title: "Telnet Service"
    - Description: "Disable the telnet service, as it can be easily exploited by attackers."
    - RuleID: 333
    - FixID: 444
    - Fix Text: "Run the command no service telnet to disable the telnet service."
    - CheckID: 555
    - Check Text: "Verify that the telnet service is disabled before granting access."

**Step 3: Generate 'Best\_Practice\_Report\_Findings.csv'**

Here is the CSV file containing the findings:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mistake Description** | **Severity** | **Title** | **Description** | **RuleID** | **FixID** | **Fix Text** | **CheckID** | **Check Text** |
| Incorrect password for administrative account | High | Password Complexity | The password should be at least 12 characters long, contain uppercase letters, numbers, and special characters. | 123 | 456 | Change the password to meet the recommended requirements. | 789 | Verify that the password meets the recommended requirements before granting access. |
| Unsecured SSH protocol version | Medium | SSH Protocol Version | Update the SSH protocol version to a more recent version (e.g., SSHv2). | 901 | 111 | Run the command ssh -V to check the current SSH protocol version and update it if necessary. | 222 | Verify that the SSH protocol version is updated before granting access. |
| Insecure telnet service enabled | High | Telnet Service | Disable the telnet service, as it can be easily exploited by attackers. | 333 | 444 | Run the command no service telnet to disable the telnet service. | 555 | Verify that the telnet service is disabled before granting access. |

**Step 4: Generate 'Best\_Practice\_Report.docx'**

Here is the Word document containing the detailed findings:

**Introduction**

This report provides a comprehensive analysis of the network configuration and identifies several mistakes against best practices. The report includes recommendations for improving the network configuration based on industry standards.

**Summary of Mistakes**

The following mistakes were identified in the network configuration:

* Incorrect password for administrative account
* Unsecured SSH protocol version
* Insecure telnet service enabled

**Detailed Findings**

**Mistake 1: Incorrect Password for Administrative Account**

* Description: The current password is not strong enough and does not meet the recommended requirements.
* Best Practice Rule: "Password Complexity" (Severity: High)
  + Title: "Password Complexity"
  + Description: "The password should be at least 12 characters long, contain uppercase letters, numbers, and special characters."
  + Fix Text: "Change the password to meet the recommended requirements."
* Explanation: The current password is not secure and can be easily guessed or cracked by attackers. It is recommended to change the password to a strong, unique value that meets the recommended requirements.

**Mistake 2: Unsecured SSH Protocol Version**

* Description: The current SSH protocol version is not secure and should be updated to a more recent version (e.g., SSHv2).
* Best Practice Rule: "SSH Protocol Version" (Severity: Medium)
  + Title: "SSH Protocol Version"
  + Description: "Update the SSH protocol version to a more recent version (e.g., SSHv2)."
  + Fix Text: "Run the command ssh -V to check the current SSH protocol version and update it if necessary."
* Explanation: The current SSH protocol version is not secure and can be exploited by attackers. It is recommended to update the SSH protocol version to a more recent version (e.g., SSHv2) to ensure security.

**Mistake 3: Insecure Telnet Service Enabled**

* Description: The telnet service is enabled, which can be easily exploited by attackers.
* Best Practice Rule: "Telnet Service" (Severity: High)
  + Title: "Telnet Service"
  + Description: "Disable the telnet service, as it can be easily exploited by attackers."
  + Fix Text: "Run the command no service telnet to disable the telnet service."
* Explanation: The telnet service is an insecure protocol that should be disabled. It is recommended to run the command no service telnet to disable the telnet service and prevent potential security risks.

**Conclusion**

This report provides a comprehensive analysis of the network configuration and identifies several mistakes against best practices. Recommendations are provided for improving the network configuration based on industry standards. It is essential to address these issues promptly to ensure the security and integrity of the network.

Please let me know if you need any further assistance or clarification.

1. Extract as PDF:

A screenshot of a computer

AI-generated content may be incorrect.

# Further Actions

1. Enhancing the pipeline to Extract other file formats other than just the in-chat text and the PDF version (at least .csv & .docx files).
2. Currently, the output is very accurate and fulfills the requirements but the model may be improved if a newer or more accurate LLM is employed (which would take massive disc space but easily implemented in the pipeline through Ollama or the API KEYs integrations).
3. Testing the Pipeline with another model rather than llama 3.2 (such as Deepseek R2 or llama 3.3, etc.).
4. Testing the same prompts used with llama 3.2 model implementation with the OpenAI API integrated flow and comparing the accuracy and reliability of the local (offline) solution against hosted (online) one.

# Thank you for this exciting opportunity to apply my knowledge and skills in LLM, RAG, and Prompt engineering domains.

1. Hopefully, it is just the beginning of great achievements and practical growth moving forward 😊
2. All the best, Rami Shoula.