

```

(base) └─[roland@Rolands-MacBook-Pro] - [~/code/Nur] - [Sat Dec 02, 02:05]
└─[$] <git:(main*)> ls -R
LICENSE                                __pycache__                          confluence_integration
crhoma                                main.py                             requirements.txt
utility                               configuration.py                     credentials.py
README.md                             old                                 setup
database

./__pycache__:
configuration.cpython-38.pyc credentials.cpython-38.pyc

./confluence_integration:
__init__.py                          context                              old
page_content.json retrieve_space.py

./confluence_integration/context:

./confluence_integration/old:
comments_summary.py page_content.json test_confluence.py

./crhoma:
__init__.py                          chroma.py                           chroma_db
path_to_persist_directory

./crhoma/chroma_db:
chroma.sqlite3

./crhoma/path_to_persist_directory:
chroma.sqlite3

./database:
__init__.py                          __pycache__                          clear_database.sh
confluence_data.db confluence_database.py

./database/__pycache__:
__init__.cpython-38.pyc
confluence_database.cpython-38.pyc

./old:
Dockerfile                          consumer.py                          docker-compose.yml
test_pulsar.py

./setup:
__init__.py                          create_db.sh                         requirements.txt setup_and_run.sh

./utility:
__init__.py
(base) └─[roland@Rolands-MacBook-Pro] - [~/code/Nur] - [Sat Dec 02, 02:05]
└─[$] <git:(main*)> cat ./setup/setup_and_run.sh
#!/bin/bash

# Function to check and install Miniconda if necessary

```

```

check_miniconda() {
    if ! [ -x "$(command -v conda)" ]; then
        echo "Miniconda is not installed. Installing Miniconda..."
        wget https://repo.anaconda.com/miniconda/Miniconda3-latest-
Linux-x86_64.sh -O miniconda.sh
        bash miniconda.sh -b -p $HOME/miniconda
        export PATH="$HOME/miniconda/bin:$PATH"
        echo "Miniconda installed."
    else
        echo "Miniconda is already installed."
    fi
}

# Determine the project root path
current_dir="$(basename "$PWD")"
parent_dir="$(basename "$(dirname "$PWD")")"

if [ "$current_dir" = "setup" ] && [ "$parent_dir" = "Nur" ]; then
    # In /Nur/setup, navigate to /Nur and set project root path
    project_root_path="$(dirname "$PWD")"
    echo "In /Nur/setup. Project root is: $project_root_path"
    cd ..
elif [ "$current_dir" = "Nur" ]; then
    # Already in /Nur, set project root path
    project_root_path="$PWD"
    echo "Already in /Nur. Project root is: $project_root_path"
else
    # Not in /Nur or /Nur/setup, clone the repo and set project root
    path
    git clone https://github.com/MDGrey33/Nur.git
    cd Nur
    project_root_path="$PWD"
    echo "Cloned repository. Project root is: $project_root_path"
fi

# Define path for setup directory
setup_path="$project_root_path/setup"

# Check if the Miniconda environment already exists
env_name="myenv"
if conda info --envs | grep -q "$env_name"; then
    echo "Miniconda environment '$env_name' already exists.
Activating it."
else
    echo "Creating Miniconda environment '$env_name'."
    conda create -n "$env_name" python=3.8 -y
fi

# Activate the Miniconda environment
# Modify this depending on your shell compatibility
source activate "$env_name" || conda activate "$env_name"

# Install Python dependencies from the setup directory
requirements_file="$setup_path/requirements.txt"

```

```

if [ -f "$requirements_file" ]; then
    echo "Installing Python dependencies from $requirements_file."
    pip install -r "$requirements_file"
else
    echo "No requirements.txt found in $setup_path. Skipping Python
dependencies installation."
fi

# Add run create_db.sh to this script
# Start the Docker containers
# echo "Starting Docker containers from $project_root_path."
# docker run -it -p 6650:6650 -p 8080:8080 --mount
source=pulsardata,target=/pulsar/data --mount
source=pulsarconf,target=/pulsar/conf apachepulsar/pulsar:3.1.1 bin/
pulsar standalone

```

```

(base) └─[roland@Rolands-MacBook-Pro] - [~/code/Nur] - [Sat Dec 02,
02:05]
└─[$] <git:(main*)> cd database
(base) └─[roland@Rolands-MacBook-Pro] - [~/code/Nur/database] - [Sat
Dec 02, 02:06]
└─[$] <git:(main*)> sqlite3 ./confluence_data.db
SQLite version 3.41.2 2023-03-22 11:56:21
Enter ".help" for usage hints.
sqlite> .schema
CREATE TABLE space_data (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    space_key TEXT,
    url TEXT,
    login TEXT,
    token TEXT
);
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE page_data (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    page_id TEXT,
    space_key TEXT,
    title TEXT,
    author TEXT,
    createdAt DATETIME,
    lastUpdated DATETIME,
    content TEXT,
    comments TEXT
);
sqlite>

```