Zenoh Installation Guide (Fully Manual Method)

```
cd ~/zenoh_project
python3 -m venv zenoh-venv
source zenoh-venv/bin/activate
```

Step 1: Install Rust and Cargo

Zenoh is built using Rust, so you must install Rust and Cargo first.

1. Install Rust and Cargo using rustup:
   ```
   curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
   ```
   When prompted, press 1 to install Rust with the default settings.

2. Reload Rust after installation:
   ```
   source $HOME/.cargo/env
   ```

3. Verify Rust and Cargo installation:
   ```
   rustc --version
   cargo --version
   ```
   If both commands return version numbers, Rust and Cargo are installed correctly.

Step 2: Add Cargo to Your System PATH

Ensure your system detects Cargo by adding it to your PATH:

```
export PATH="$HOME/.cargo/bin:$PATH"
echo 'export PATH="$HOME/.cargo/bin:$PATH"' >> ~/.zshrc
source ~/.zshrc
```

Verify that Cargo is accessible:

```
cargo --version
```

Step 3: Create a Python Virtual Environment

Using a virtual environment prevents dependency conflicts.

1. Navigate to your project directory:
   ```
   cd /Users/azizahalq/Desktop/project2/zenoh-python
   ```

2. Create a new virtual environment:
   ```
   python3.10 -m venv zenoh_env
   ```

3. Activate the virtual environment:
   ```
   source zenoh_env/bin/activate
   ```

4. Verify Python is using the virtual environment:
   ```
   which python3
   ```

Step 4: Install Dependencies

Upgrade package manager and install required dependencies:

```
pip install --upgrade pip setuptools wheel maturin
brew install cmake ninja
```

Step 5: Clone and Build Zenoh

1. Delete any existing Zenoh directory:
   ```
   rm -rf zenoh-python
   ```

2. Clone the Zenoh repository:
   ```
   git clone https://github.com/eclipse-zenoh/zenoh-python.git
   ```

```
   cd zenoh-python
```

3. Build Zenoh using Cargo:
```
   cargo build --release
```

Step 6: Install Zenoh in Python
After successfully building Zenoh, install it inside your virtual environment:
```
maturin develop
```

Step 7: Verify Zenoh Installation
Test the Zenoh installation:
```
python3 -c "import zenoh; print('Zenoh installed successfully!')"
```

If you see "Zenoh installed successfully!", then Zenoh is installed correctly.

Step 8: Fix Python Module Path Issues (If Needed)
If Zenoh is installed but Python cannot find it, manually set the PYTHONPATH:

```
export PYTHONPATH=$PYTHONPATH:/Users/azizahalq/Desktop/project2/zenoh-python/zenoh_env/lib/python3.10/site-packag
```

Then, save it permanently:
```
echo 'export PYTHONPATH=$PYTHONPATH:/Users/azizahalq/Desktop/project2/zenoh-python/zenoh_env/lib/python3.10/site-p
source ~/.zshrc
```

Step 9: Test Zenoh Pub-Sub Communication

Subscriber (Receiver)
Run this script to listen for messages:
```
import zenoh

session = zenoh.open()

def callback(sample):
    print(f"Received: {sample.payload.decode()}")

sub = session.declare_subscriber("test/topic", callback)
print("Listening for messages on 'test/topic'...")
input("Press Enter to exit...\n")
```

Publisher (Sender)
Run this script in another terminal to send messages:
```
import zenoh
import time

session = zenoh.open()
pub = session.declare_publisher("test/topic")

for i in range(5):
    pub.put(f"Hello Zenoh! Message {i+1}")
    print(f"Sent: Hello Zenoh! Message {i+1}")
    time.sleep(1)
```

Final Checklist
 Rust and Cargo installed (rustc --version, cargo --version)

Cargo added to system PATH (export PATH="$HOME/.cargo/bin:$PATH")
Python virtual environment created and activated
Dependencies installed (pip install maturin, brew install cmake ninja)
Zenoh manually cloned and built (cargo build --release)
Zenoh installed in Python (maturin develop)
Zenoh successfully imported in Python (import zenoh)
Zenoh pub-sub tested successfully

Zenoh is now fully installed and working! Let me know if you need help integrating it into your project.