# **Name: Abdurrahman Qureshi**

# **Roll No: 242466**

Practical No: 3

Date Of Performance: 22/01/2024

Aim: To install and configure the NS2 simulator on Windows 10 using Ubuntu on WSL (Windows Subsystem for Linux), and to create and run a basic simulation script to demonstrate its functionality.

Lab Objectives:

To install NS2 on Windows 10 using WSL and Ubuntu, understand its

architecture with C++ and OTcl, and create a basic simulation to explore NS2's

capabilities for network research.

Lab Outcomes:

Successfully install NS2, gain hands-on experience with OTcl scripting and

C++ protocol implementation, and design simple network topologies to

simulate and analyze network behavior using trace and NAM files.

What is NS2?

**NS2 (Network Simulator 2)** is a discrete event network simulator widely used for research and educational purposes in computer networking. It provides a platform for simulating various types of networks and protocols, including wired and wireless networks. NS2 allows users to model network behaviors, test new network protocols, and analyze the performance of networks under different conditions.

Features:

* Simulation of Network Protocols: Supports protocols like TCP, UDP, routing protocols, multicast, etc.
* Customizability: Users can modify or create new network protocols.
* Support for Wired and Wireless Networks: Simulates a wide range of network types, including ad-hoc and mobile networks.
* Use of OTcl and C++:
  + OTcl (Object Tool Command Language): Used for configuring and controlling simulations.
  + C++: Used for implementing the underlying mechanisms and protocols.
* Visualization: Results can be visualized using tools like NAM (Network Animator) for better analysis.

Applications:

* Research in networking protocols and performance evaluation.
* Teaching network concepts in academic environments.
* Experimentation with network designs before real-world deployment.

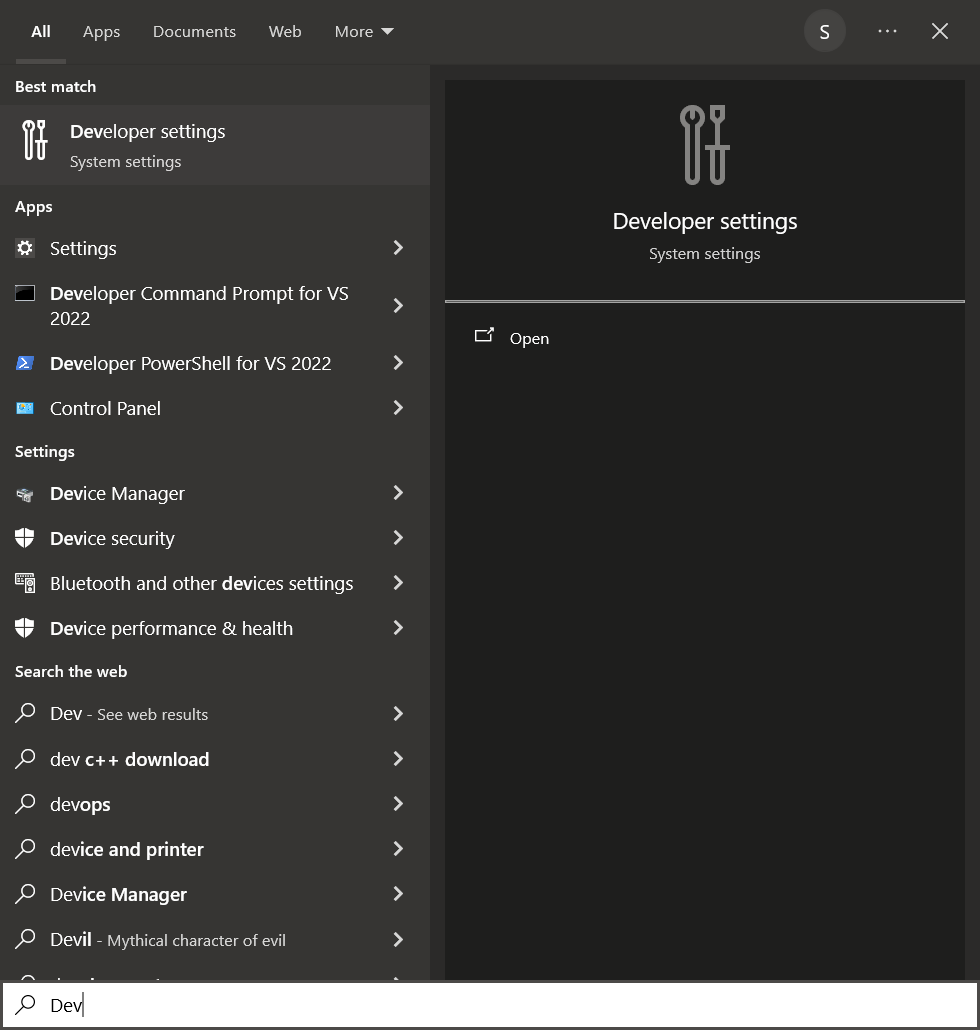
Prerequisites:

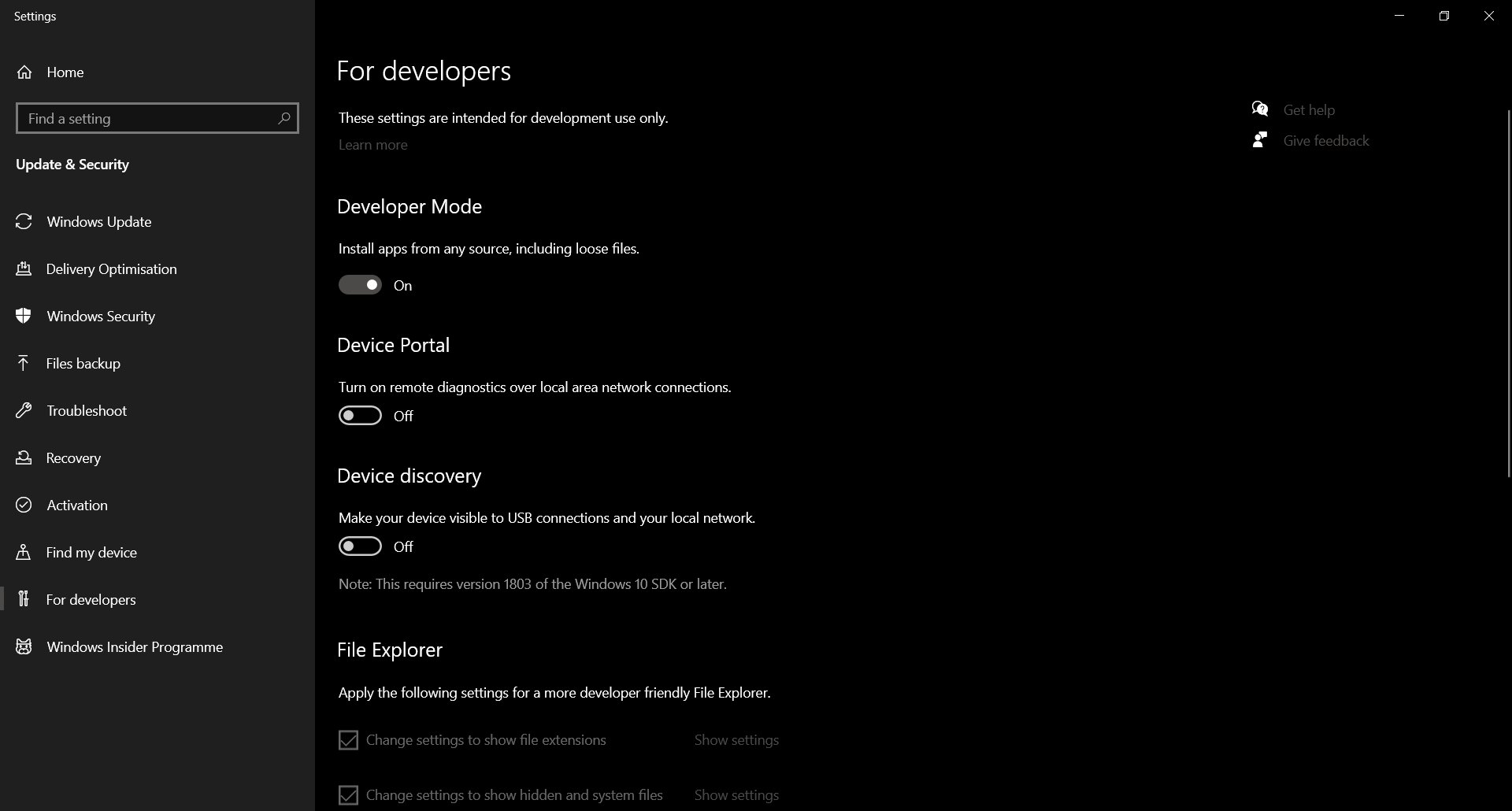
* Windows 10 Operating System
* Administrator privileges for installing software
* Internet connection to download required software packages

Steps to setup:

Step 1: Enable Developer Mode in Windows 10

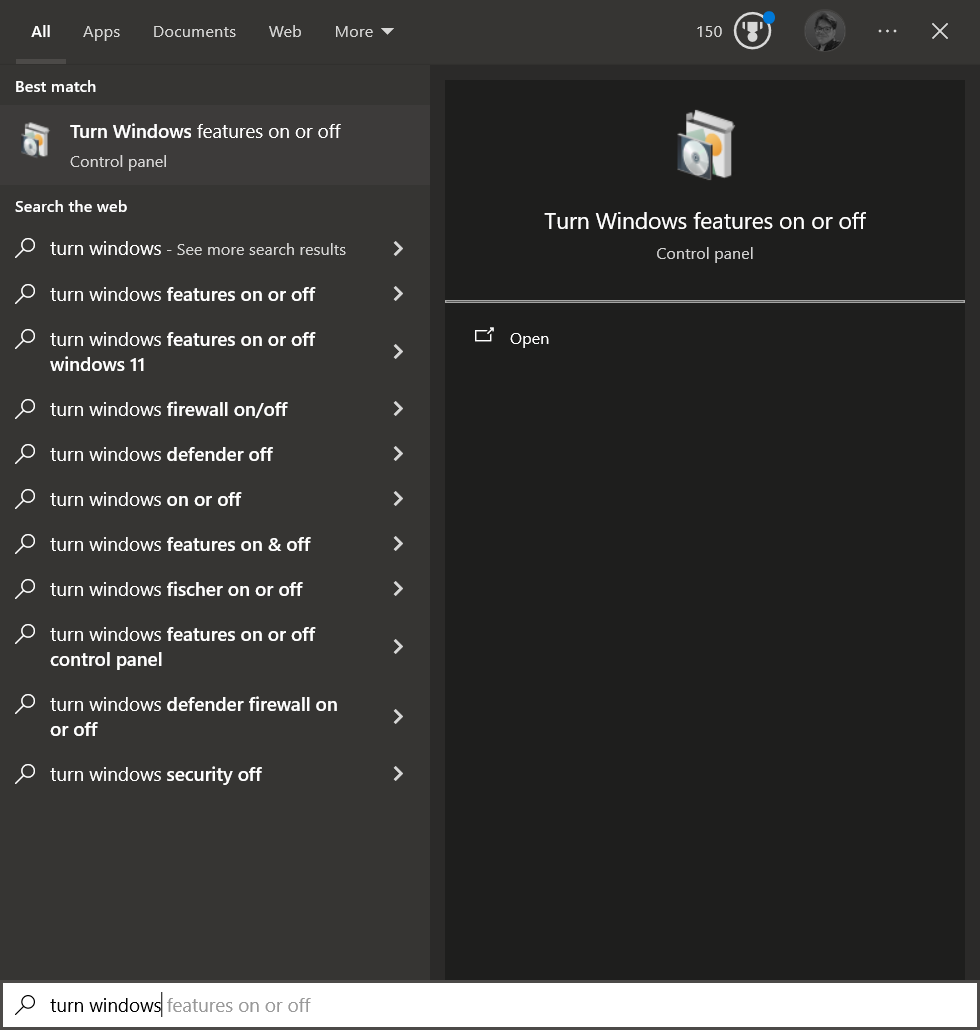
* Press Windows Key and search for "Developer settings".
* Open "Developer settings" and toggle on "Install apps from any source". Confirm by clicking YES.

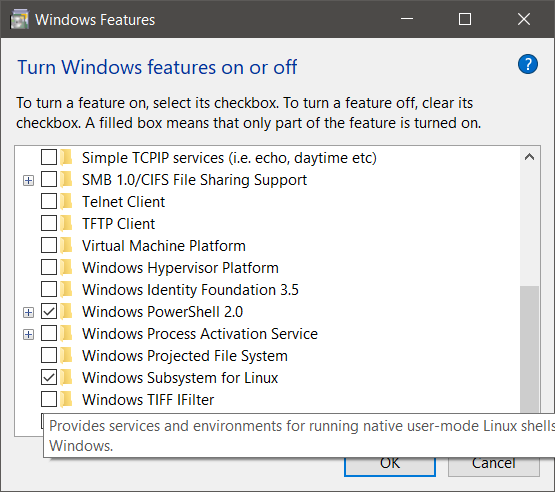




Step 2: Enable Windows Subsystem for Linux (WSL)

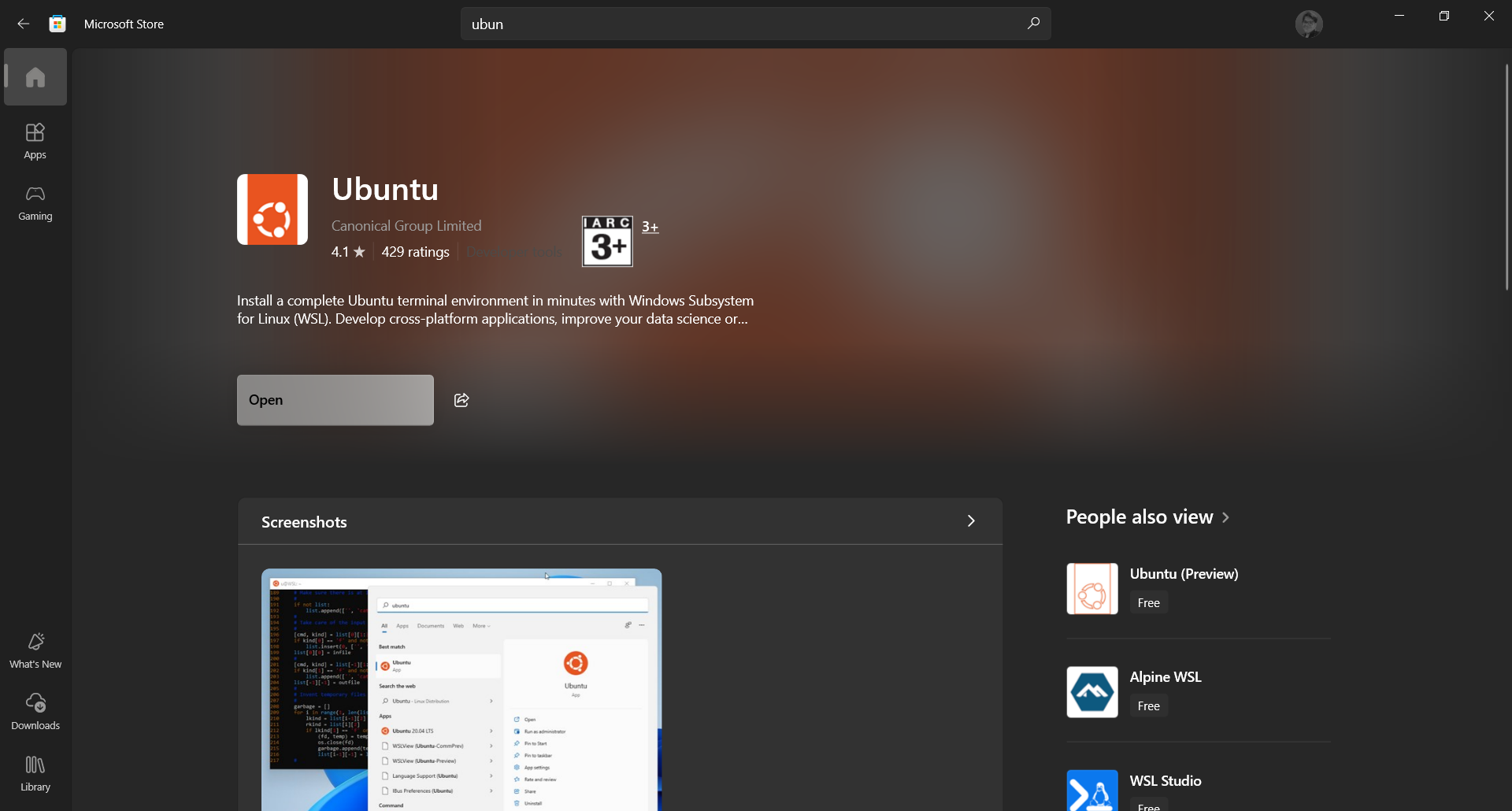
* Press Windows Key, search for "Windows Features", and click on "Turn
* Windows features on or off".
* Scroll down and check the box for "Windows Subsystem for Linux".
* Click OK and restart your computer





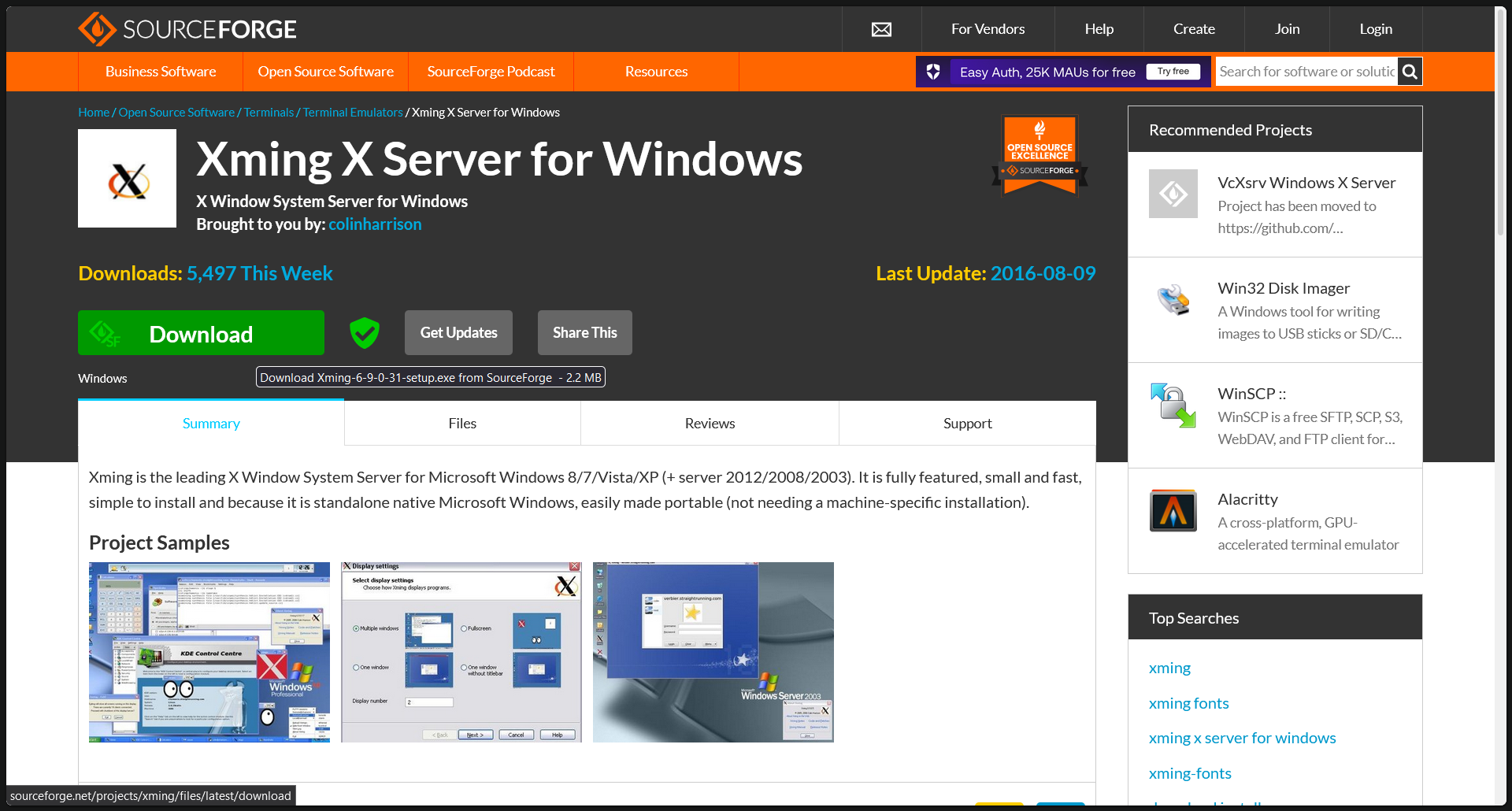
Step 3: Install Ubuntu from the Microsoft Store

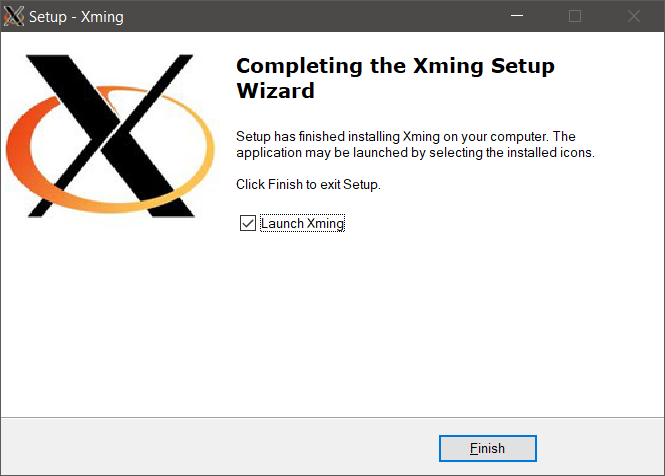
* Open Microsoft Store by pressing Windows + S and typing "Microsoft Store".
* Search for "Ubuntu" and click Install to download Ubuntu (approximately 700 MB).



Step 4: Install Xming for Graphical Interface

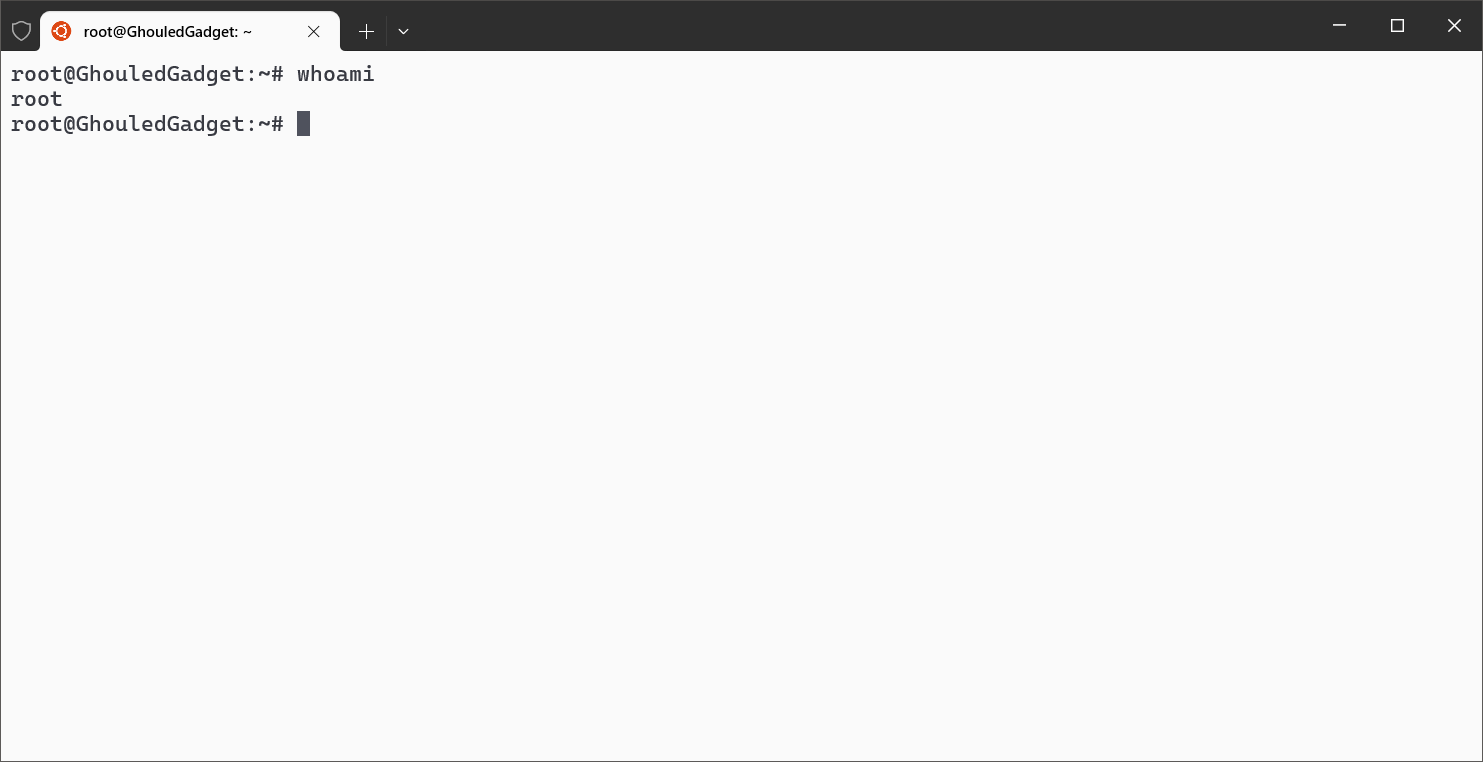
* Download Xming from SourceForge.
* Install Xming by following the prompts.
* After installation, open Xming (you should see the Xming icon in the system tray).





Step 5: Install NS2 and Required Packages

* Open the Ubuntu application from the Start Menu.
* Set up a username and password for the Ubuntu environment

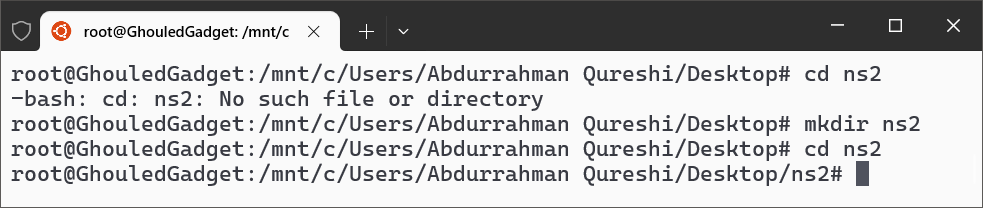


Step 6: Create a Working Directory

Change to a working directory where you want to save your NS2 files.

For example:

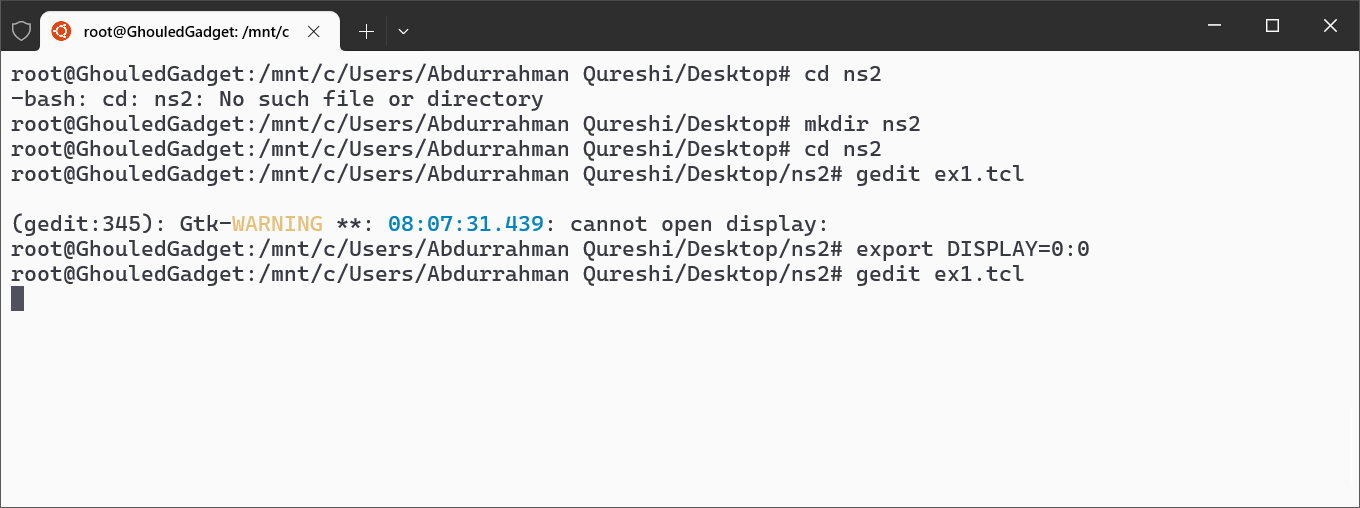
cd /mnt/c/Users/Admin/Desktop/ns2

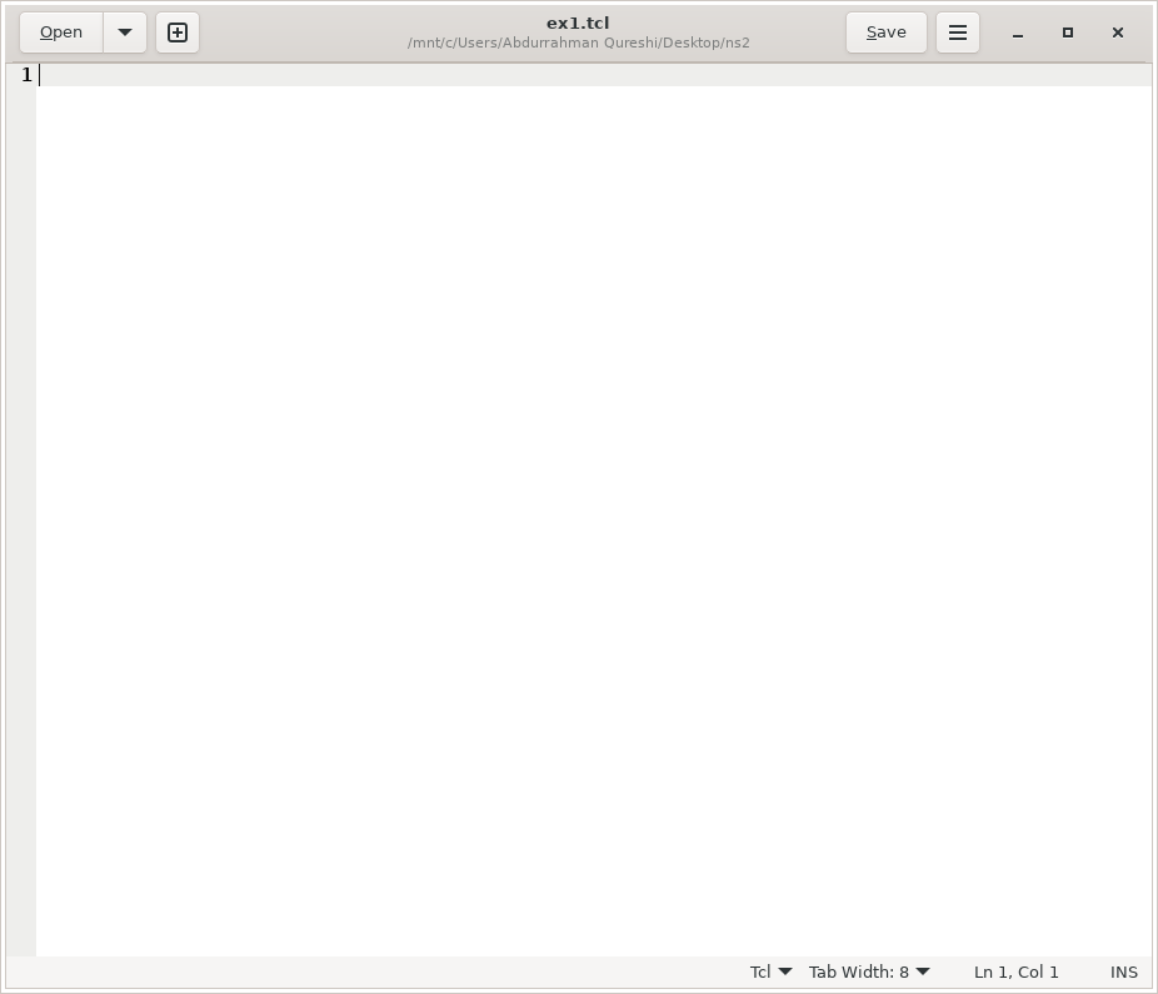


Step 7: Open gedit and Create a Simulation Script

Open gedit by typing the following command in the terminal:

gedit ex1.tcl





* In the gedit window, write the following sample NS2 script:

# Create global variables

set ns [new Simulator]

# Set nam trace

set namf [open wired1.nam w]

$ns namtrace-all $namf

# Open the trace file

set tracef [open wired1.tr w]

$ns trace-all $tracef

# Creating nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

# Establish links between nodes

$ns duplex-link $n0 $n1 2Mb 100ms DropTail

$ns duplex-link $n1 $n2 200Kb 100ms DropTail

# Label nodes

$ns at 0.0 "$n0 label Client1"

$ns at 0.0 "$n1 label Server"

$ns at 0.0 "$n2 label Client2"

# Node colors

$n0 color blue

$n1 color red

$n2 color green

# Procedure to finish simulation

proc finish {} {

global ns tracef namf

$ns flush-trace

close $tracef

close $namf

exec nam wired1.nam &

exit 0

}

# Schedule finish procedure

$ns at 2.0 "finish"

$ns run

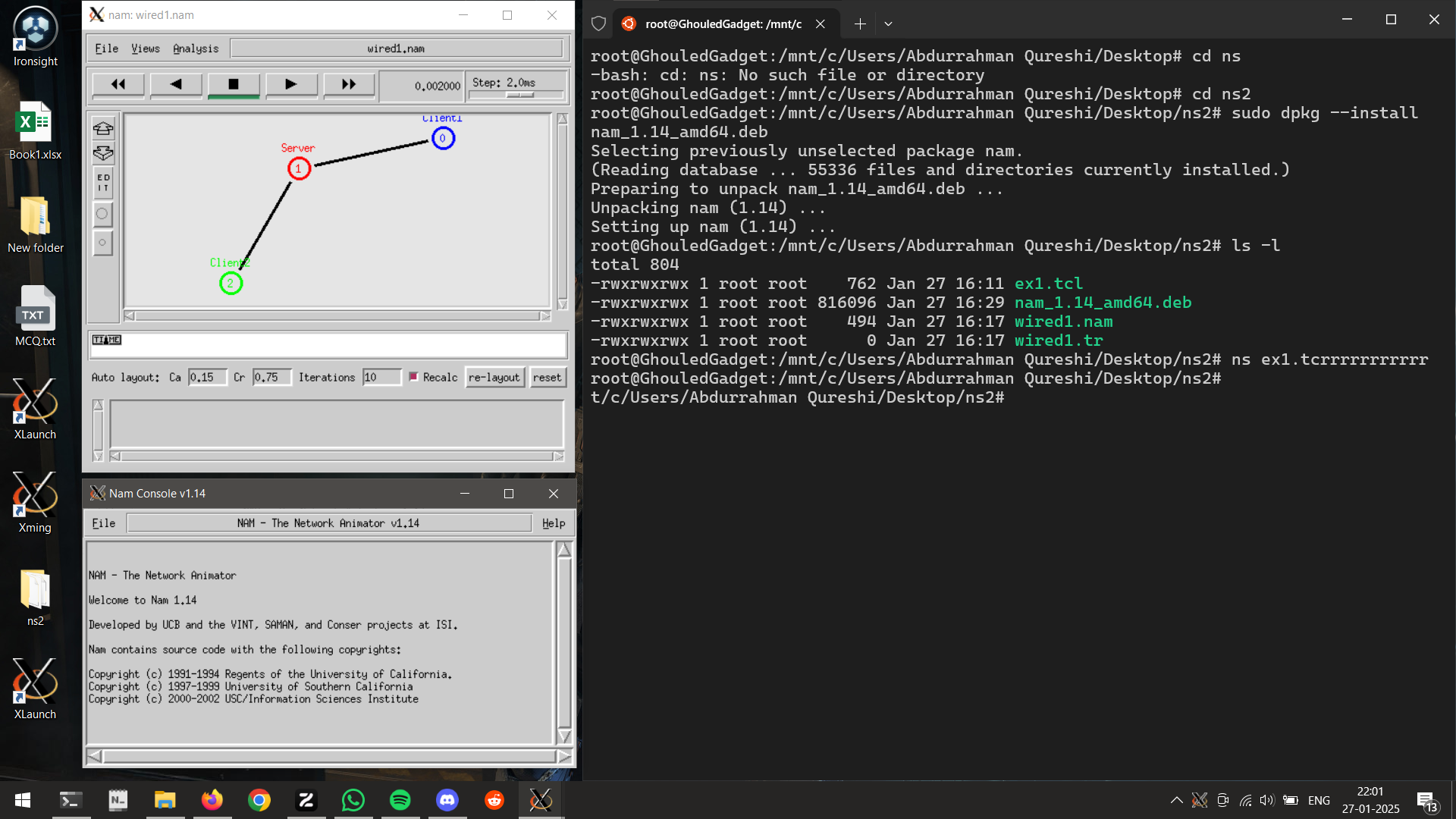
* Save the file as "ex1.tcl" and close gedit.

Step 8: Run the Simulation

* In the terminal, run the simulation by typing:

ns ex1.tcl

* The simulation will run, and you will see the NAM window displaying the network topology.



Understanding the NS2 Script:

* $ns: Represents the NS2 simulator instance.
* Simulator setup: Defines the simulation environment and generates a NAM file (wired1.nam) for visualization and a trace file (wired1.tr) for analysis.
* Node creation and configuration: Creates nodes (n0, n1, n2) and establishes duplex links between them.
* Labeling and coloring nodes: Labels the nodes and sets their colors for better visualization.
* Simulation control: Defines a procedure (finish) that will stop the simulation after 2 seconds, close the trace files, and launch the NAM visualization tool.

Conclusion:

During this practical session, we installed and set up NS2 on Windows 10 using Ubuntu on WSL. We created a basic simulation script in OTcl to design a simple network topology, ran the simulation, and visualized the results using NAM. This hands-on activity helped us understand the functionality of NS2, the role of OTcl in configuring simulations, and how to use terminal commands to interact with NS2.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Performance  (7M) | Journal  (3M) | Lab Ethics  (2M) | Attendance  (3M) | Total  (15M) | Faculty Signature |
|  |  |  |  |  |  |