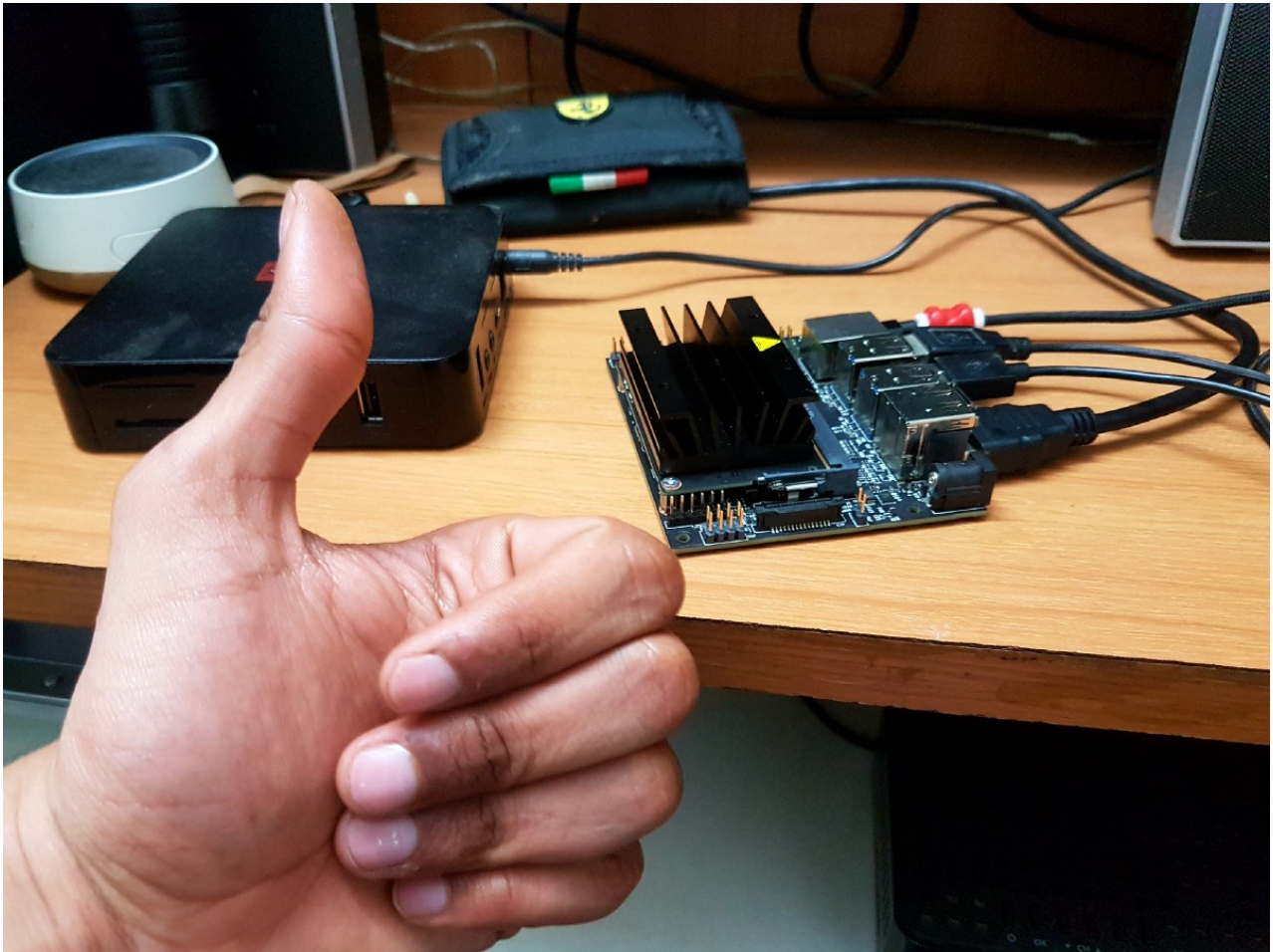# Jetson Nano -Remote VNC Access

medium.com/@bharathsudharsan023/jetson-nano-remote-vnc-access-d1e71c82492b

Bharath Sudharsan   February 3, 2020

[Bharath Sudharsan](#)

Feb 3, 2020

.

4 min read

In this story, I will run you through the steps to connect to Jetson Nano via Remmina or other VNC viewer application. By default, when we open the Desktop Sharing application which is pre-installed in Jetson Nano, it crashes. After following the steps from this story, you can avoid those crashes.

Step 1: Connecting the Board to Your Wireless Network

It turns out the NVIDIA L4T has poor support for USB Wi-Fi adaptors, and most of the adaptors don't work with the distribution. So buy a compatible WIFI USB adapter to connect to your local network.

Step 2: Enabling Desktop Sharing

Unfortunately, the instructions helpfully left on the Jetson's desktop on how to enable the installed VNC server from the command line don't work, and going ahead and opening the Settings application on the desktop and clicking on "Desktop Sharing" also fails as the Settings app silently crashes. A problem that appears to be down to an incompatibility with the older Gnome desktop. **I will tell you the easiest route to get through and make the Desktop Sharing application work normal.**

First, open the terminal and paste this line.

```
sudo nano /usr/share/glib-2.0/schemas/org.gnome.Vino.gschema.xml
```

Next, add (paste) the following key into the XML file which just opened in nano editor.

```
<key name='enabled' type='b'>
<summary>Enable remote access to the desktop</summary>
<description>
If true, allows remote access to the desktop via the RFB
protocol. Users on remote machines may then connect to the
desktop using a VNC viewer.
</description>
<default>true</default>
</key>
```

adding the key in the XML file

Now, after pasting, save and exit. Next, Compile the schemas for Gnome using the command below

```
sudo glib-compile-schemas /usr/share/glib-2.0/schemas
```

Now, at this stage, the "Desktop Sharing" panel crashing should have stopped. So open the Desktop Sharing app and

Tick the "*Allow other users to view your desktop*" and also "*Allow other users to control your desktop*" checkmarks. Then make sure "*You must confirm each access to this machine*" is turned off. Finally tick the "*Require the user to enter this password*" checkmark, and enter a password for the VNC session.

Use these settings in the Desktop sharing app

Step 3: Start the VNC server on every startup

Open 'startup applications' using the search box that appears at the top of the screen.

Now, click `Add` at the right of the box, then type 'Vino' in the name box, and then in the command box enter `/usr/lib/vino/vino-server`. Click `Save` at the bottom right of the box, and then close the app.

Next, we need to disable encryption of the VNC connection to get things working. To do this, open the terminal and enter the following commands

```
$ gsettings set org.gnome.Vino require-encryption false
$ gsettings set org.gnome.Vino prompt-enabled false
```

You can now go ahead and reboot the board, once rebooted, log back into your account. VNC should now be running and serving the desktop.

Step 4: Connect to your Nano via VNC

Find the IP of your Jetson nano board using '*ifconfig*' command. I found mine in the wlan0 as 192.168.0.106

Use this command to check whether your VNC server is running or not.

```
ps -ef|grep vnc
```

vnc server is running

Now, once you know your IP and also confirm the VNC server is up and running, its time to connect to your Jetson Nano via any VNC client app.

Open Remmina click the add button on the top left, then select VNC as the protocol, then click the (…)button which is in the right end of the server IP address entering bar. This (…) button should automatically scan and show the name of your jetson Nano. Now just click, enter your password and connect.

The (…) button

Scan results showing my Jetson nano, click on it to connect

Step 5: Enable auto-login and Disable / Turn Off Lock Screen (optional)

The VNC server starts only after the user logs in. In an ideal headless setup, we need a keyboard and monitor to log in, doing this is at every login in tiring. So I usually enable auto-login and disable the lock screen.

Open the Activities overview and start typing Users.

> Click Users to open the panel, select the user account that you want to log in to automatically at startup, Press Unlock in the top right corner and type in your password when prompted, Switch the Automatic Login switch to on.

To Disable / Turn Off Lock Screen please refer this link

https://websiteforstudents.com/disable-turn-off-ubuntu-18-04-lts-beta-lock-screen/