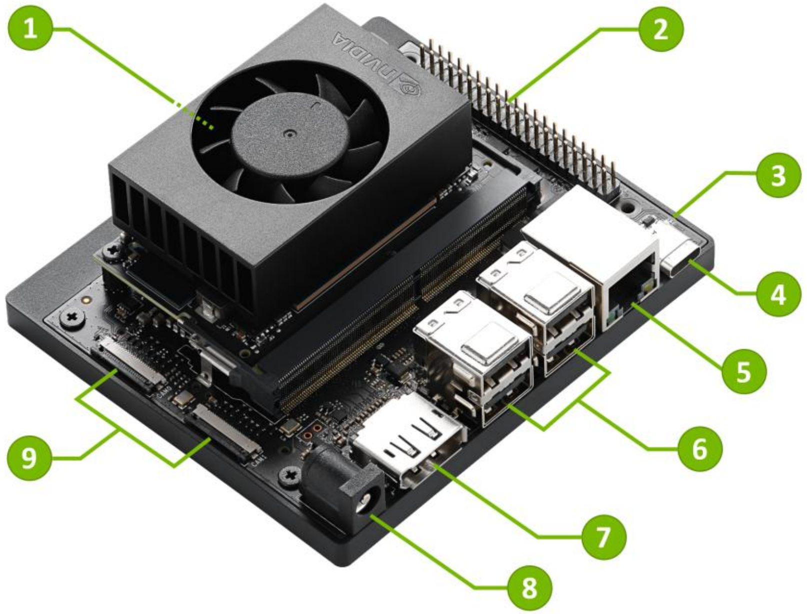
Jetson Orin Nano Setup & Linux Basics – Get hands-on with the hardware



NIELIT Chandigarh/Ropar

"AI at the edge. Power in your pocket. Jetson Nano."

Jetson Orin Nano Setup & Linux Basics

- Next-gen AI edge device by NVIDIA
- Designed for advanced robotics, computer vision & AI at the edge
- Key features:
    - 5–40 TOPS of AI performance
    - Orin architecture (Ampere GPU + ARM Cortex-A78AE)

Jetson Orin Nano Developer Kit Specs

Component	Specification
CPU	6-core Arm Cortex-A78AE v8.2 64-bit
GPU	1024-core Ampere with 32 Tensor Cores
RAM	8 GB LPDDR5
Storage	microSD card slot
Ports	USB 3.0, HDMI, Ethernet, GPIO, CSI
Power	5V/4A via DC barrel jack or USB-C

Use Cases

- Edge AI
- Robotics
- Smart cameras
- Industrial automation
- AIoT (AI + IoT)

What's in the Box

- Jetson Orin Nano Dev Kit
- Quick Start Guide
- Heatsink/fan attached
- microSD card slot (no card included)
- Power supply (optional)
- HDMI cable, keyboard, mouse – BYO

Initial Hardware Setup

1. Insert a flashed microSD card (JetPack Image)
2. Connect monitor via HDMI
3. Attach keyboard, mouse via USB
4. Connect to power (DC barrel or USB-C)
5. Wait for first boot (Linux setup)

Flashing JetPack to SD Card

- Download image: developer.nvidia.com
- Use **Balena Etcher** or **Raspberry Pi Imager**
- Flash JetPack image to SD card (32GB+ recommended)
- Insert card into Jetson and boot

First Boot Setup

- Select language, timezone, username
- Connect to Wi-Fi or Ethernet
- Update system:

```
sudo apt update && sudo apt upgrade
```

- Jetson Linux is Ubuntu-based (22.04 or later)

Jetson SDK Components

- JetPack includes:
  - CUDA
  - cuDNN
  - TensorRT
  - OpenCV
  - DeepStream SDK
- Preinstalled or install via SDK Manager or apt

Jetson Orin Nano vs Jetson Nano

Feature	Orin Nano	Jetson Nano
AI Performance	Up to 40 TOPS	~0.5 TFLOPS
RAM	8 GB LPDDR5	4 GB LPDDR4
GPU	Ampere + Tensor	Maxwell (128-core)
Price	~$199	~$99

Linux Terminal Basics

- Open terminal:

Ctrl + Alt + T

- Key commands:

```
pwd       # print working directory
ls        # list files
cd        # change directory
mkdir     # make directory
touch     # create file
rm        # remove file
```

Editing Files on Jetson

- Use nano (easy):

```
nano myscript.py
```

Or vim, gedit, VS Code (remote)

Installing Packages

- Python packages (pip):

```
sudo apt install python3-pip
pip3 install numpy opencv-python
```

- System packages (apt):
- sudo apt install git htop cmake

Writing & Running Python Code

```
nano hello.py
print("Hello from Jetson Orin Nano!")
```

- Run it:
- python3 hello.py

Checking GPU Support

```
nvidia-smi
```

- If unsupported on Jetson, use:

```
tegrastats
```

- Check TensorRT, CUDA availability:

```
python3 -c "import torch;
print(torch.cuda.is_available())"
```

Sample AI Project (Demo)

- Run object detection using pre-trained YOLO or SSD models
- Use Jetson-inference repo
- Setup:

```
git clone https://github.com/dusty-nv/jetson-inference

cd jetson-inference

./build.sh
```

Best Practices

- Use heatsink/fan for cooling
- Monitor CPU/GPU temps with tegrastats
- Don’t write to SD excessively
- Use external SSD for storage if needed

Troubleshooting Tips

- No display? Check SD, HDMI cable, power supply
- Flash failed? Try Etcher again
- JetPack errors? Use latest image version
- Use dmesg, journalctl for logs