aiDAPTIV+
Pro Suite 2.0 Install guide
(For 50 Series GPU only)

Preliminary 0.3

Phison Electronics Corporation
Tel: +886-37-586-896 Fax: +886-37-587-868
E-mail: sales@phison.com / suppport@phison.com

Phison may make changes to specifications and product description at any time without notice. PHISON and the Phison logo are trademarks of Phison Electronics Corporation, registered in the United States and other countries. Products and specifications discussed herein are for reference purposes only. Copies of documents which include information of part number or ordering number, or other materials may be obtained by emailing us at sales@phison.com or support@phison.com.

REVISION HISTORY

Revision	Draft Date	History	Pro Suite Version	Author
0.1	2025/06/30	Preliminary release	NOUN_2.0.6T (beta version)	Sean Liou
0.2	2025/07/08	Update section 1.1	NOUN_2.0.6T (beta version)	Sean Liou
0.3	2025/07/14	Update section 1.1, 2.2 and 2.4	NOUN_2.0.6T (beta version)	Sean Liou

TABLE OF CONTENTS

LIST OF FIGURES

1. ENVIRONMENT PREPARATION

This program provides one-click installation.
Users only need to execute **install.sh** to complete the deployment.

Name	Version
Pro Suite	NOUN_2.0.6T
aiDAPTIVLink	aidaptiv-vNXUN_2_04_00
Supported GPU	5060Ti、5070Ti, 5080

1.1. Supported OS and Nvidia driver version

Category	Detail
OS	Ubuntu 22.04 LTS Desktop
GPU driver	Linux-x86_64-570.133.07、CUDA 12.8
Kernel	Version 6.8 or later
gcc	Version 12.3

1.2. Precautions before installation

1. Please do not use special characters in folder or file names
such as whitespace, , !, ?, ', @, #, $, %, ^, &, *, (,), +

2. TCP Port

Service Name	Port
Pro Suite Web	8899, 8799
Pro Suite Service	3019, 5432, 9400, 7017, 8000
Prometheus	3090, 9100

3. Confirm that the installation environment can connect to the **external network**.
4. If you are using SSH to remotely connect to the machine, please enter the following command enable X11 forwarding.

```
ssh -X user@remote_host
```

5. Please enter the following command in the terminal to confirm whether the system recognizes the NVIDIA GPU card.

```
user@prosuite-dev:~$ sudo lshw -c display
[sudo] password for user:
  *-display
       description: VGA compatible controller
       product: NVIDIA Corporation
       vendor: NVIDIA Corporation
       physical id: 0
```

6. Please enter the following command in the terminal to confirm the current time zone of the system.

```
# Check timezone on linux
user@prosuite-dev:~$ cat /etc/timezone
Asia/Taipei
```

7. To change the time zone, please enter the following command in the terminal

```
# Change timezone, e.g to Asia/Taipei
user@prosuite-dev:~$ sudo timedatectl set-timezone Asia/Taipei
```

2. INSTALLATION

2.1. Update Kernel version

1. Use the command *uname -r* to check the kernel version.

If it is not version 6.8 or newer, please proceed with the following steps to update the kernel. If it is already version 6.8 or newer, please skip to step 2.2.

2. If you have installed the kernel version using mainline, it may cause unexpected errors. You need to remove the mainline and 060800 kernel packages first (skip this step if not applicable).

```
$ sudo dpkg -l | grep "mainline\|060800"
$ sudo apt purge -y {kernel package name}
```

3. Stop all docker container

```
$ sudo docker ps -a
$ sudo docker rm -f {container name}
```

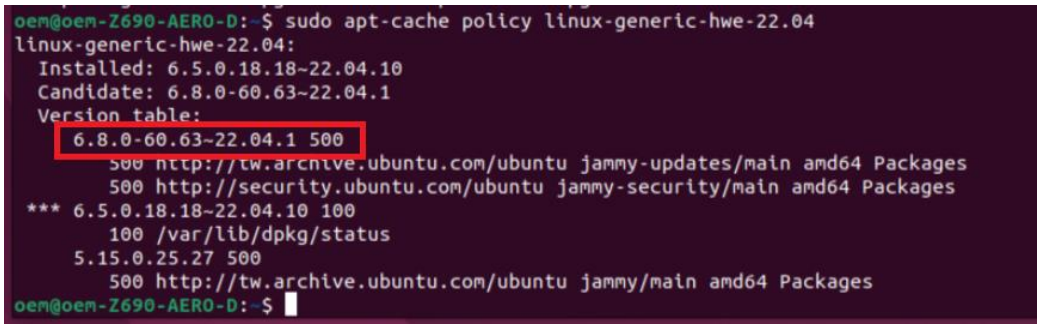
4. Update OS kernel, it will take a few minutes

Updating Kernel Source

```
$ sudo apt update
```

When executing the command, make sure that version 6.8.0 is displayed. If it is not shown, it means that sudo apt update did not succeed.

```
$ sudo apt-cache policy linux-generic-hwe-22.04
$ sudo apt install -y linux-generic-hwe-22.04
$ sudo update-grub
$ sudo update-initramfs -u
$ sudo reboot
```



Updating Kernel Dependency Packages

```
$ sudo apt upgrade
```

Installing Necessary Linux Packages

```
$ sudo apt install build-essential linux-headers-$(uname -r)
```

2.2. GCC-12 Installation

Enter the following command to install GCC version 12.3

```
$ sudo apt install gcc-12                  #Install gcc version 12
$ sudo rm /usr/bin/gcc                     #Delete the link of the old version of gcc
$ sudo ln -s /usr/bin/gcc-12 /usr/bin/gcc  #Set the gcc default path to point to version
12
```

Confirm the gcc version through the following command:

```
$ gcc –version
```

```
root@oem:/opt/NOUN_2.0.6T-0003# gcc --version
gcc (Ubuntu 12.3.0-1ubuntu1~22.04) 12.3.0
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

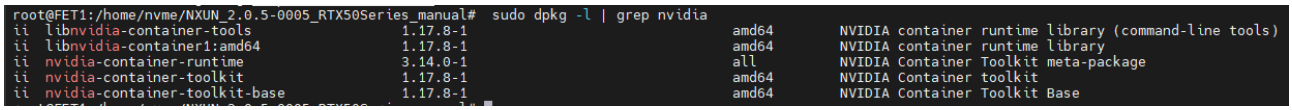
Use the following command to confirm that the gcc path has been linked to version 12

```
$ ll /usr/bin/gcc
```

```
root@oem:/opt/NOUN_2.0.6T-0003# ll /usr/bin/gcc
lrwxrwxrwx 1 root root 15  七  10 14:13 /usr/bin/gcc -> /usr/bin/gcc-12*
```

2.3. Install NVIDIA Driver (Linux-x86_64-570.133.07)

1. Confirm that there are no NVIDIA packages in dpkg. If there are, they need to be removed.




```
$ sudo dpkg -l | grep nvidia

$ sudo apt-get remove --purge '^nvidia-.*'

$ sudo apt-get remove --purge '^libnvidia-.*'
```

2. Remove the NVIDIA packages installed using the ***.run installer***.

```
$ sudo nvidia-uninstall
```

3. You can use the following command to check if any related processes are running:

```
$ sudo lsof /dev/nvidia*
```

4. If there are processes using the NVIDIA driver, you need to stop these processes.

For example, stop the X server

```
$ sudo systemctl stop display-manager #If the system crashes when using this command,
please log out of the GUI and press Ctrl+Alt+F2 to switch to text mode to install.
```

5. Or stop CUDA programs and the NVIDIA persistence daemon:

```
$ sudo systemctl stop nvidia-persistenced
```

6. Use the following command to uninstall the NVIDIA driver:

```
$ sudo modprobe -r nvidia-drm

$ sudo modprobe -r nvidia
```

7. Check if any .so files are in use, which could cause the .run file installation to fail

   → The .run file will create symbolic links under **/usr** . If there are existing files, it will cause the installation to fail.

```
$ sudo find /usr -name libnvidia*   # remove all directories found
```

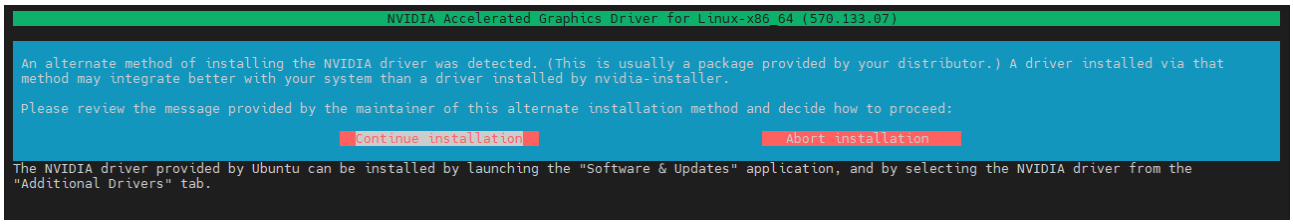

```
$ rm -r /user/lib/x86_64-linux-gpu/libnvidia-ml.so.1/
```

8. Change to the Prosuite installation package directory and use the .run file within the package.

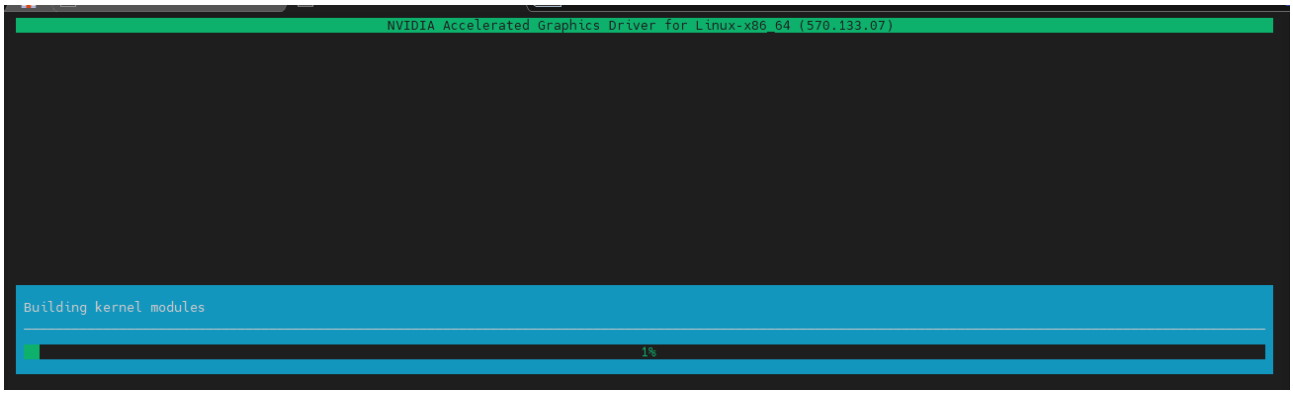
```
root@FET1:/home/nvme/NXUN_2.0.5-0005_RTX50Series_manual# ls -ll
total 367080
drwxrwxr-x 8 nvme nvme      4096  三  17 18:53 config
-rwxrwxr-x 1 nvme nvme      3595  三  17 18:53 export_log.sh
-rwxrwxr-x 1 nvme nvme     69881  四  11 17:06 install.sh
-rw-rw-r-- 1 nvme nvme       120  三  17 18:53 LICENSE
-rwxrwxr-x 1 nvme nvme 375774798  四  11 16:24 NVIDIA-Linux-x86_64-570.133.07.run
-rwxrwxrwx 1 nvme nvme      6820  三  17 18:53 phison-compose.yaml
-rwxrwxrwx 1 nvme nvme      1361  六  18 17:51 phison-config.env
-rwxrwxrwx 1 nvme nvme       886  六  18 17:18 phison-image.env
drwxrwxr-x 4 nvme nvme      4096  三  17 18:53 pkg
-rw-rw-r-- 1 nvme nvme        83  三  17 18:53 README.md
```

9. Execute the command `sudo ./NVIDIA-Linux-x86_64-570.133.07.run -m=kernel-open`

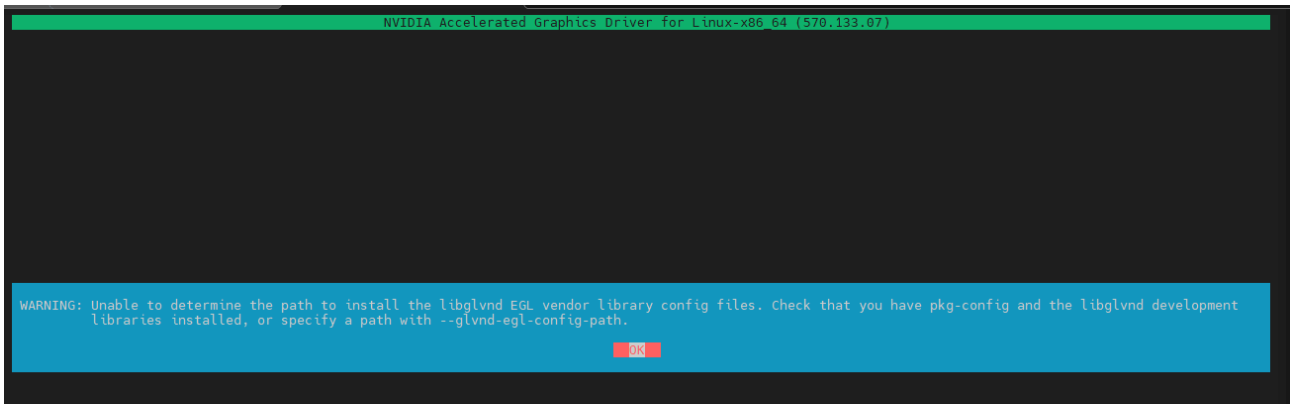
   a. An installation window will appear; click “Continue installation.”



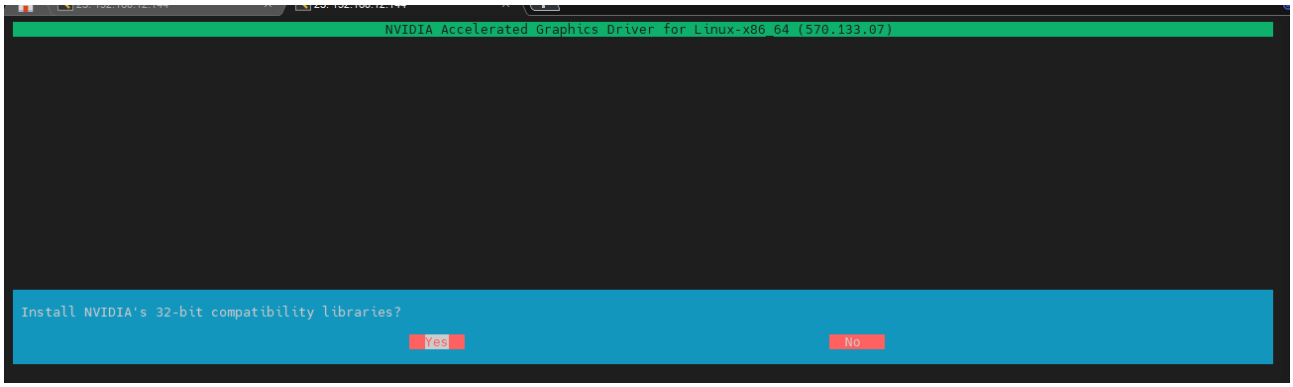
   b. The installation will start.



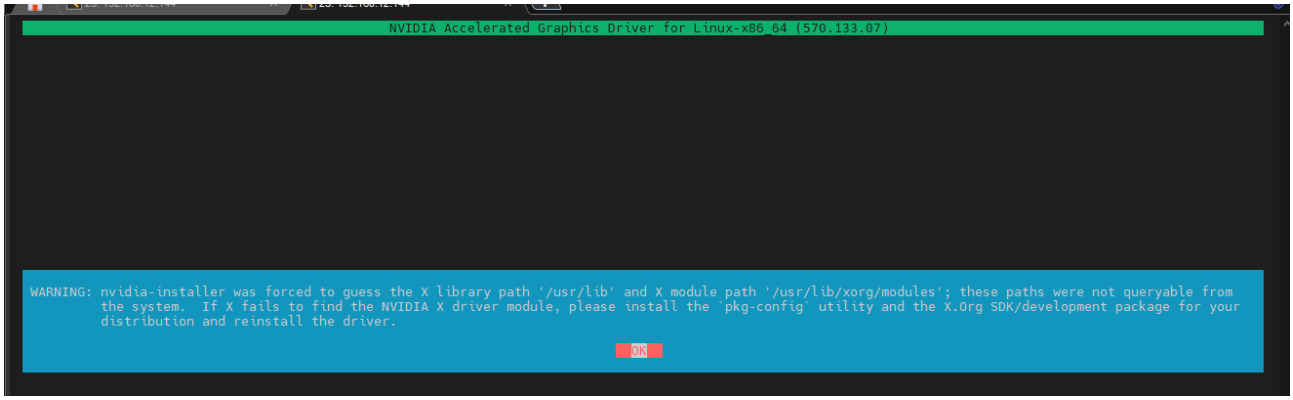
   c. Click “OK” to continue the process.



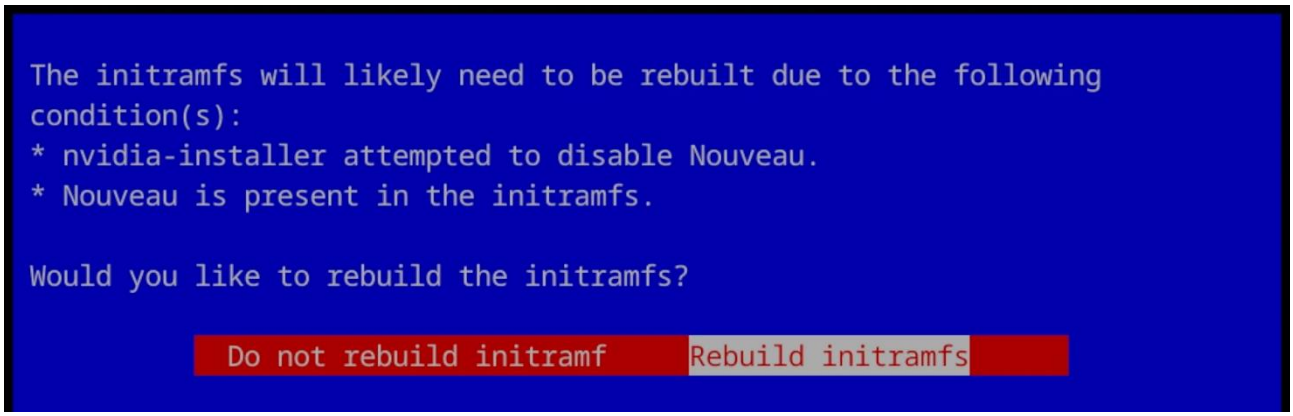
d. When asked if you want to install compatibility libraries, select “Yes.”



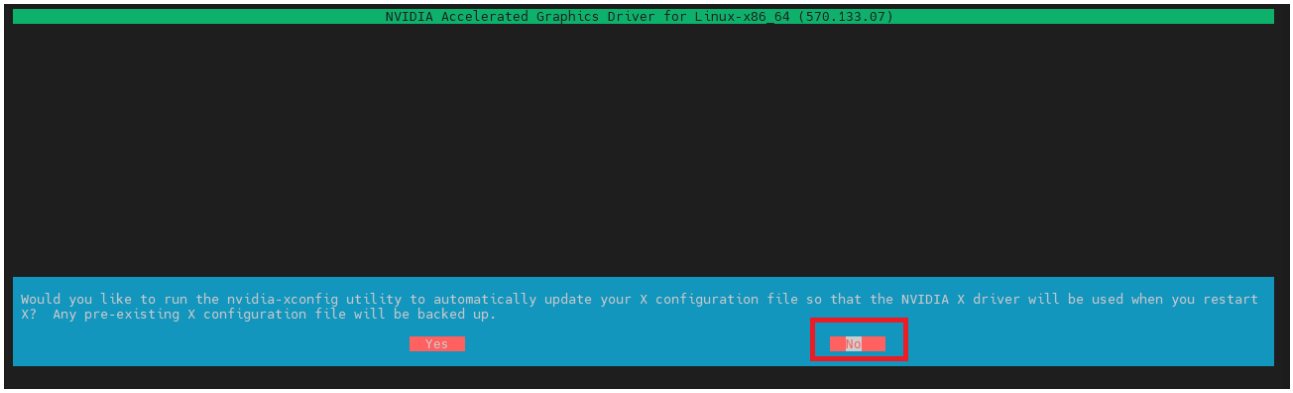
e. Click “OK” to continue the process.



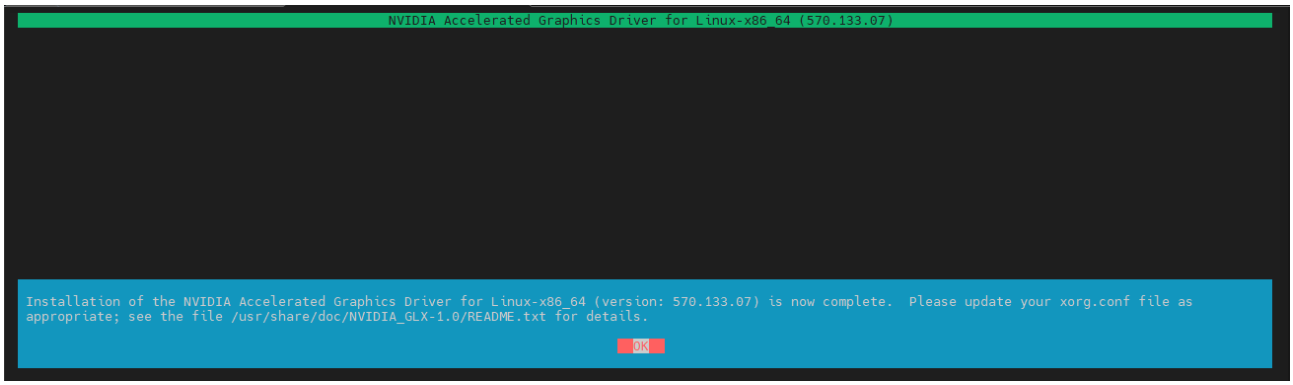
f. If the system asks whether to rebuild initramfs, please rebuild (may not appear)



g. When asked if you want to run the X utility, select “No.”



h. The installation completion message will look like the following image



i. After installation, restart Linux to make the GPU driver effective.

10. Verify the NVIDIA driver version using the command: *nvidia-smi*

```
root@ai-MS-7D48:/home/ai# nvidia-smi
Tue Apr 22 11:04:55 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 570.133.07             Driver Version: 570.133.07     CUDA Version: 12.8     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|-----------------------------------------+------------------------+----------------------|
```

2.4. Install Pro Suite

2.4.1. Precaution

- This version of Pro Suite is in offline installation mode. After completing section 2.1 to 2.3, you **CAN** perform the installation process of Pro Suite in an environment where you cannot connect to the Internet or have no network.
- When installing in an environment that cannot connect to the Internet, please first annotate apt sources to avoid installing the Linux package and trying to connect.
    - Rename apt sources

```
mv /etc/apt/sources.list /etc/apt/sources.list.bak
```

    - After completing the Pro Suite installation, change the file name back

```
mv /etc/apt/sources.list.bak /etc/apt/sources.list
```

2.4.2. Unzip Pro Suite tar file

Enter the following command in the terminal to decompress the Pro Suite installation package.

```
$ tar xvf {install package name}.tar
```

2.4.3. Execute install.sh

Enter the following command in the terminal to execute install.sh

```
$ cd {install package name}
sudo ./install.sh -c
```

2.4.4. Num 0, Install Pro Suite

Note: If LVM already exists, you need to remove LVM before installing Pro Suite.
(Please refer to *appendix A.1*)

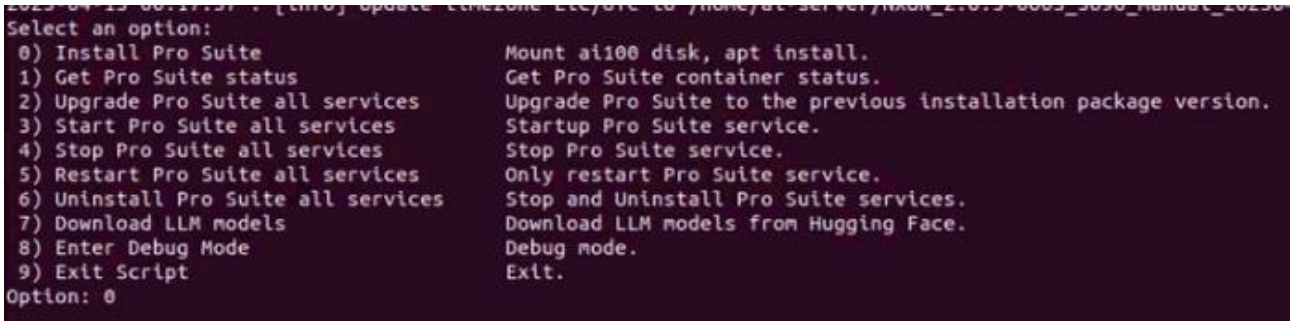
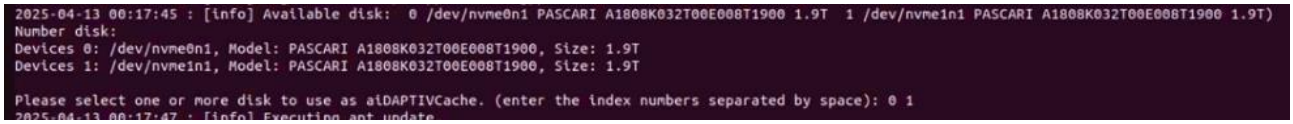


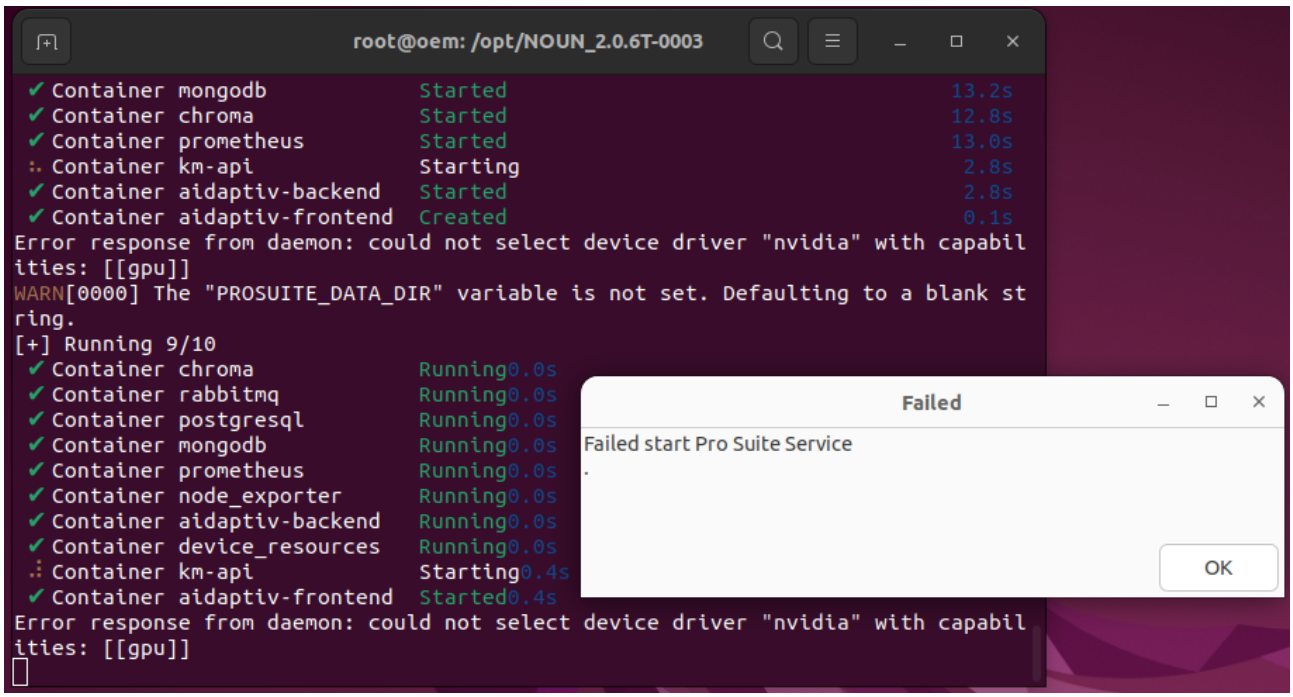
Figure 2-1 Num 0, Install Pro Suite

- Select aiDAPTIVCache



- After the installation is complete, Error response from daemon: *could not select device driver "nvidia" with capabilities: [[gpu]]* will appear. Please restart Docker and re-read the driver settings.

```
sudo systemctl restart docker
```



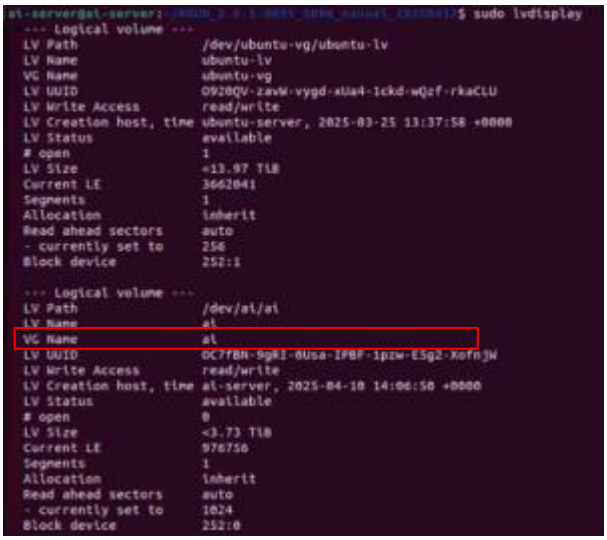
APPENDIX A – HANDLING UNEXPECTED ISSUES

A.1. aiDAPTIVCache been occupied

Here are the troubleshooting steps for the issue of aiDAPTIVCache being occupied during installation:

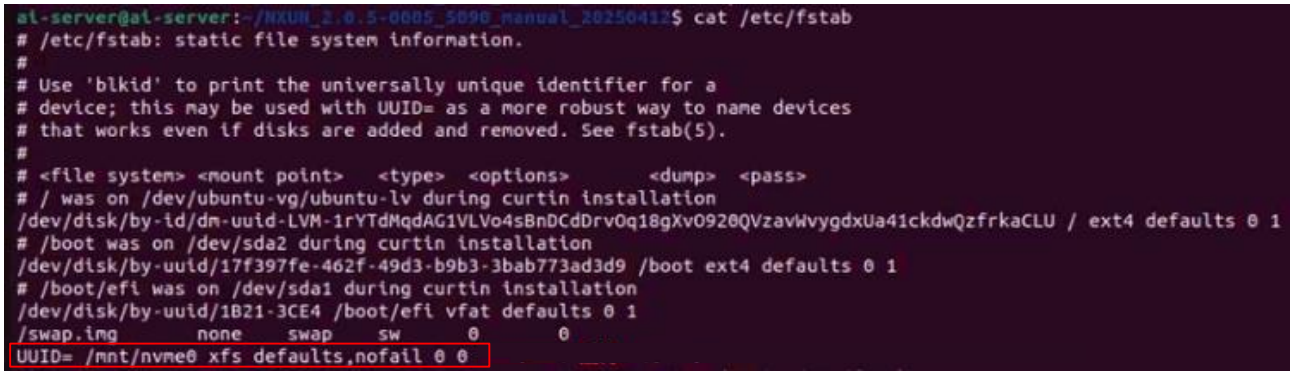
1. Check VG Name:

Normally, after aiDAPTIVCache is configured into RAID0 via Pro Suite, it will be displayed as `/dev/prosuite-vg/prosuite-rd`. If the VG name appears as something else, it indicates that someone might have manually configured the RAID.



2. Check Partition: Verify if aiDAPTIVCache has been written to the boot partition. If it has, you need to delete it.

Execute command `vim /etc/fstab`, and **DELETE** the last line *UUID= /mnt/nvme0 xfs defaults,nofail 0 0* and save.



3. Remove RAID from LVM:

If the RAID has already been written into LVM, you need to remove it. In Linux, you can use the lvremove command to remove the logical volume.

Execute command `sudo lvremove /dev/ai/ai`

A.2. Permission issue

If you encounter the error “download is performed unsandboxed as root as file… couldn’t be accessed by user _apt”, you can solve this problem in the following two ways:

- Method 1: Add a new sandbox configuration file and let apt run it with root privileges.

```
$ sudo vi /etc/apt/apt.conf.d/10sandbox
$ APT::Sandbox::User "root";
```

- Method 2: Manually install using dpkg

```
$ sudo dpkg -i {file name}.deb
```

A.3. Ubuntu auto update issue

Due to the fact that the NVIDIA 50 series GPUs are relatively new products, automatic updates in the Ubuntu operating system might lead to mismatches between the Kernel, GPU driver, and CUDA versions. Here are the steps to check and troubleshoot this issue:

1. Check Ubuntu Automatic Update Logs: Review the logs to see what updates have been applied automatically.

```
cat /var/log/dpkg.log
```

2. Reinstall the NVIDIA Driver:
   - If the environment **CAN** connect to the internet, it is recommended to first run `sudo apt update && sudo apt upgrade -y` to ensure all packages are up to date, and then proceed to install the NVIDIA driver.
   - If the environment **CANNOT** connect to the internet, you can directly reinstall the NVIDIA driver (refer to *section 2.2*).

3. Disable Ubuntu Automatic Update Function:
   You can disable the automatic update feature in Ubuntu. However, please evaluate the security risks associated with this action.

```
$ sudo vim /etc/apt/apt.conf.d/50unattended-upgrades
```

Uncomment the first two lines of Allowed-Origins{} and then save.

```
Unattended-Upgrade::Allowed-Origins {
//      "${distro_id}:${distro_codename}";
//      "${distro_id}:${distro_codename}-security";
        // Extended Security Maintenance; doesn't necessarily exist for
        // every release and this system may not have it installed, but if
        // available, the policy for updates is such that unattended-upgrades
        // should also install from here by default.
        "${distro_id}ESMApps:${distro_codename}-apps-security";
        "${distro_id}ESM:${distro_codename}-infra-security";
//      "${distro_id}:${distro_codename}-updates";
//      "${distro_id}:${distro_codename}-proposed";
//      "${distro_id}:${distro_codename}-backports";
```

A.4. Error response from daemon: could not select device driver “nvidia” with capabilities: [[gpu]]

1. Check the driver information

Run `nvidia-smi` to check if the driver information is displayed successfully. If not, it indicates that the installation has failed and needs to be reinstalled.

```
root@ws2050-vv4:/home/hm_chuang# nvidia-smi
Tue Jul  1 11:01:39 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.107.02             Driver Version: 550.107.02     CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
```

2. Ensure that the runtime is installed

Run `nvidia-container-runtime --version` to ensure that the runtime is installed. If installed, it will display the version information as shown below; if not installed, you will see **nvidia-container-runtime: command not found**.

```
root@ws2050-vv4:/home/hm_chuang# nvidia-container-runtime --version
NVIDIA Container Runtime version 1.16.1
commit: a470818ba7d9166be282cd0039dd2fc9b0a34d73
spec: 1.2.0

runc version 1.1.12
commit: v1.1.12-0-g51d5e94
spec: 1.0.2-dev
go: go1.21.9
libseccomp: 2.5.4
```

3. Check if nvidia-container-runtime is set up

Enter the command `cat /etc/docker/daemon.json` to check if nvidia-container-runtime is set up. If it is, the message within the red box will appear.

```
root@FET1:/home/nvme/NXUN_2.0.5-0005_RTX50Series_manual# cat /etc/docker/daemon.json
{
    "default-runtime": "nvidia",
    "exec-opts": [
        "native.cgroupdriver=systemd"
    ],
    "runtimes": {
        "nvidia": {
            "args": [],
            "path": "/usr/bin/nvidia-container-runtime"
        }
    },
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "100m",
        "max-file": "3"
    },
    "storage-driver": "overlay2",
    "insecure-registries":["harbor.phison.com"]
}
```

4. Restart Docker: Use the following command to restart Docker.

```
sudo systemctl restart docker
```