Headless Device Mode

For this course, we are running the Jetson Nano Developer Kit in a **"headless"** configuration. That means you do not hook up a monitor directly to the Jetson Nano Developer Kit. This method conserves memory resources on the Jetson Nano and has the added benefit of eliminating the requirement for extra hardware, i.e. a monitor, keyboard, and mouse.

In addition, we will further simplify the configuration by using **"USB Device Mode"**. In this mode, your Jetson Nano Developer Kit connects directly to your computer through a USB cable. This eliminates the need for a network connection on the Jetson Nano, as well as the need to determine the IP address on your network. It is always `192.168.55.1` in this mode.

In the the steps that follow, you'll boot your Nano and log on from your computer with SSH protocol, a common method for communicating with embedded systems like the Jetson Nano. The course notebooks are included in a Docker container environment, which will be downloaded from the Internet and stored on your microSD card.

Setup Steps

1. Connect your flashed JetPack microSD card and power supply (DC barrel jack for 4G kit, USBC connector for 2G kit). The Jetson Nano Developer Kit will power on and boot automatically.
2. A green LED next to the Micro-USB connector will light as soon as the developer kit powers on. Wait about 30 seconds. Then connect the USB cable from the Micro USB port on the Jetson Nano Developer Kit to the USB port on your computer.
3. Connect your USB camera to a USB port on the Jetson Nano. *(If using the alternate CSI camera, connect it to the CSI port.)*
4. If you are downloading the Docker container with the course notebooks in it for the first time, connect your Nano to the Internet using the Ethernet port or a compatible WiFi device.
5. On your computer, open a terminal window if using Mac or Linux, and a PowerShell window if using Windows. In the terminal, log on to the Jetson Nano with the following command, where `<username>` is the values you set up on your Nano during the operating system configuration:

   ```
   ssh <username>@192.168.55.1
   ```

Enter the password you configured when asked.

6. Add a data directory for the course with the following command in the Jetson Nano terminal you've logged into:
   `mkdir -p ~/nvdli-data`
7. Run the Docker container with the following command, where `<tag>` is a combination of the course version and Jetson Nano JetPack L4T operating system version (form is `<tag>` = `<course_version>-<L4T_version>`). A list of tags can be found in the course NVIDIA NGC cloud page.

```
sudo docker run --runtime nvidia -it --rm --network host \
    --volume ~/nvdli-data:/nvdli-nano/data \
    --device /dev/video0 \
    nvcr.io/nvidia/dli/dli-nano-ai:<tag>
```

To create and run a reusable script for this step try the following (example tag shown):

```
# create a reusable script
echo "sudo docker run --runtime nvidia -it --rm --network host \
    --volume ~/nvdli-data:/nvdli-nano/data \
    --device /dev/video0 \
    nvcr.io/nvidia/dli/dli-nano-ai:v2.0.2-r32.7.1" > docker_dli_run.sh

# make the script executable
chmod +x docker_dli_run.sh

# run the script
./docker_dli_run.sh
```

If using the alternate CSI camera instead of the USB webcam, add `--volume /tmp/argus_socket:/tmp/argus_socket` to your docker run command. For example:

```
# create a reusable script
echo "sudo docker run --runtime nvidia -it --rm --network host \
    --volume ~/nvdli-data:/nvdli-nano/data \
    --volume /tmp/argus_socket:/tmp/argus_socket \
    --device /dev/video0 \
    nvcr.io/nvidia/dli/dli-nano-ai:v2.0.2-r32.7.1" > docker_dli_run.sh

# make the script executable
chmod +x docker_dli_run.sh

# run the script
./docker_dli_run.sh
```

If using the alternate CSI camera AND the 2GB version, also add `--memory=500M --memory-swap=4G` to your docker run command. For example:

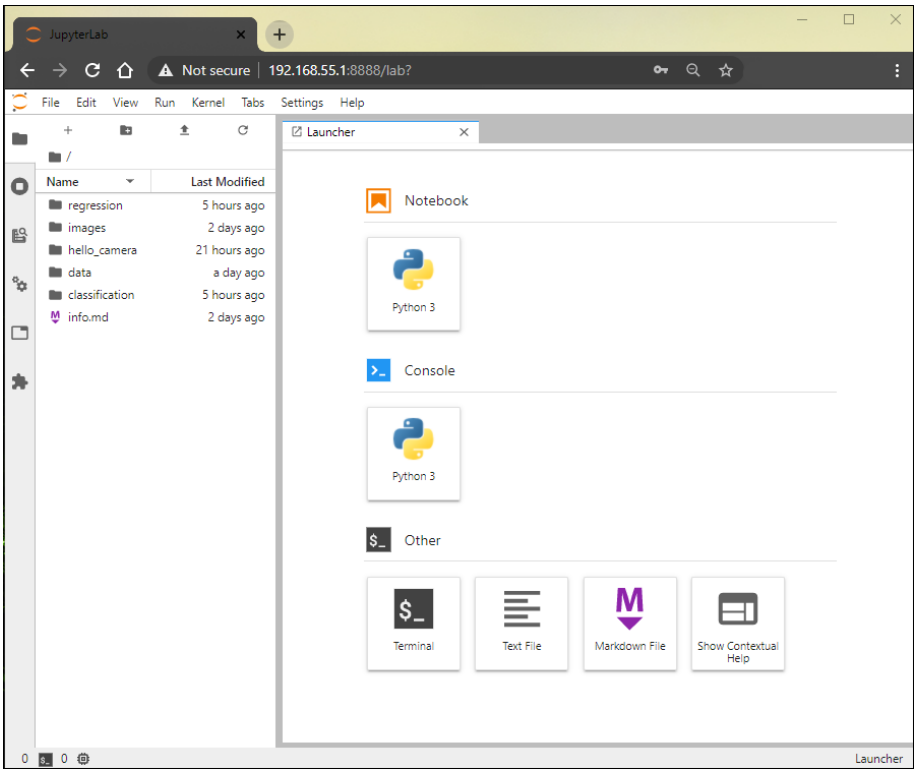
```
# create a reusable script
echo "sudo docker run --runtime nvidia -it --rm --network host \
    --memory=500M --memory-swap=4G \
    --volume ~/nvdli-data:/nvdli-nano/data \
    --volume /tmp/argus_socket:/tmp/argus_socket \
    --device /dev/video0 \
    nvcr.io/nvidia/dli/dli-nano-ai:v2.0.2-r32.7.1" > docker_dli_run.sh
```

```
# make the script executable
chmod +x docker_dli_run.sh

# run the script
./docker_dli_run.sh
```

Logging Into The JupyterLab Server

1. Open the following link address : 192.168.55.1:8888. The JupyterLab server running on the Jetson Nano will open up with a login prompt the first time.
2. Enter the password: `dlinano`. You will see this screen. Congratulations!



Troubleshooting

- **The LED does not light up when the DC barrel jack power supply is connected.**
  - *Check to be sure you have shorted the two pins on the J48 header with a jumper.*
- **The LED lights up, but I cannot access the JupyterLab server from my browser.**
  - *Try a different browser.*
- **I cannot access the JupyterLab server from any browser.**
  - *Check your computer to see if any new USB devices are recognized when plugging in the USB cable to your computer*
  - *If on Windows, check "Device Manager" to see if any new device was added.*
- **My computer does not recognize Jetson Nano when connected.**
  - *Check your USB cable to see if it is enabled for data transfer.*
  - *You can test it by connecting the Micro-B end of the USB cable to some other USB peripheral such as tablet, Kindle, or other device that communicates over a USB Micro-B port.*
- **My USB cable seems good, but my computer does not recognize Jetson Nano.**
  - *Check if your Jetson Nano Developer Kit is properly booting by connecting it to a TV through an HDMI cable. See if the TV displays the NVIDIA logo when booted, and eventually displays the Ubuntu desktop.*
- **My Jetson Nano does not show anything on the TV when booting with the TV attached.**

    - *Check if you have inserted your microSD card all the way into the microSD card slot. You should hear a small click sound.*
- **The microSD card is fully inserted, but my Jetson Nano does not boot properly.**
    - *Go back to "Write Image To The MicroSD Card" to reflash your SD card.*
- **I'm using a CSI camera with the 2GB Nano and the camera freezes during training.**
    - *Release as much RAM as possible, increase the swap, and specify memory options during the docker run launch:*

```
# Disable ZRAM:
sudo systemctl disable nvzramconfig

# Prevent X-Server from starting:
sudo systemctl set-default multi-user.target

# Create additional 6GB swap file (assume 4GB already for a total of 10GB)
sudo fallocate -l 6G /mnt/6GB.swap
sudo chmod 600 /mnt/6GB.swap
sudo mkswap /mnt/6GB.swap

# Append the following line to /etc/fstab
sudo su
echo "/mnt/6GB.swap swap swap defaults 0 0" >> /etc/fstab
exit

# REBOOT!

# Check your memory and swap:
free -h

# Create a reusable script
echo "sudo docker run --runtime nvidia -it --rm --network host \
    --volume ~/nvdli-data:/nvdli-nano/data \
    --volume /tmp/argus_socket:/tmp/argus_socket \
    --device /dev/video0 \
    --memory=500M --memory-swap=8G \
    nvcr.io/nvidia/dli/dli-nano-ai:v2.0.2-r32.7.1" > docker_dli_run.sh

# Make the script executable
chmod +x docker_dli_run.sh

# Run the script
./docker_dli_run.sh
```