

Configuring Baremetal on CentOS/RHEL

OS Installation:

1. Prepare OS installation CD / USB and start the installation.
2. Usually, try with "Use full HDD space"
3. Once OS installation is complete, edit the network scripts ← Important
 - a. Goto "/etc/sysconfig/network-scripts"
 - b. Edit the file ifcfg-<networkname> (e.g. ifcfg-eth0, ifcfg-eth1)
 - c. Edit ONBOOT=yes
 - d. Save the file with wq!

Connecting remotely to Linux Server via SSH:

- For Windows, using Putty, by adding the IP address in Host Name section in putty.
- For Linux/MacOSX, enter the IP address with

```
ssh xxx.xxx.xxx.xxx -l username
```

Disabling nouveau.modeset:

First thing to do on a baremetal setup is to disable nouveau.modeset.

To disable, Open grub.conf file via command: **vim /boot/grub/grub.conf**

In grub.conf file go to scroll down to latest kernel entry and append **nouveau.modeset=0**

Save file with **wq!**

Installing Kernel-devel: Important

Driver installation in Linux requires the package **kernel-devel** be installed.

The installed kernel and installed kernel-devel versions must be installed and versions should match. If they are not matching, the driver install will fail.

Use following commands to install **kernel-devel** and assume you are root:

- a. RHEL, CentOS, Fedora;

```
#yum install kernel
#yum install kernel-devel
#reboot
```
- b. OpenSUSE:

```
#zypper in kernel-desktop
```

```
#zypper in kernel-desktop-devel
#reboot
```

Driver Installation:

1. You must install “gcc” before you install the driver
 - a. `yum list | grep gcc` ← get exact package name
 - b. `yum install <whatever package name you got in step a>`.
2. Kill X-Server
 - a. Do this through SSH
 - b. `init 3`
 - c. `kill x`
3. Change the mode of NVIDIA installer to execute
 - a. `chmod +x <NVIDIA Installer Name>`
4. Install the driver now
 - a. `./<NVIDIA Installer Name>`

Installing x11vnc

1. There are other VNC Servers available, but they create their own X-Server instance when invoked.
2. X11vnc connects to the existing X-Server session, and hence the VNC will show GPU-driven display.
3. Issues found if running some other VNC Servers:
 - a. NVIDIA X-Server settings are not available.
 - b. Unable to run NVIDIA samples
4. X11vnc is not available in “Package Manager” or “yum”.
 - a. Either download the RPM from http://rpmfind.net/linux/atrpms/sl5-x86_64/atrpms/stable/x11vnc-0.9.9-4.el5.x86_64.rpm
 - b. Or give this location in Package Manager, so that it will resolve dependencies and install all the packages.

Editing the xorg.conf file

1. Xorg.conf file is located at `/etc/X11/xorg.conf`
2. Basically this file is needed to setup NVIDIA displays (with head or headless)
3. Make sure the X server isn’t running
 - a. You can do this via “`pgrep x`”
 - b. If running, either do “`init 3`” or “`kill x`”
4. Run following command to configure the file xorg.conf automatically
 - a. `nvidia-xconfig -a`
 - b. make sure the file has “Device”, “Monitor” and “Screen” sections.
 - c. If the server’s `/etc/X11/xorg.conf` does not have all of the GPU devices, they can be manually added. First you need to find all of the PCI Bus ID

```
# lspci | grep NVIDIA
06:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
07:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
08:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
09:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
85:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
86:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
89:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
8a:00.0 VGA compatible controller: NVIDIA Corporation GK104GL [GRID K520] (rev a1)
```

5. Now, use “nano” or “vim” to edit this file for our use

a. Delete all monitor sections, and add following monitor section

```
Section "Monitor"
Identifier      "Headless"
HorizSync       80.0 - 80.0
VertRefresh     75.0
Modeline        "1280x1024_75.00" 138.45 1280 1368 1504 1728 1024 1025 1028 1069 -
Hsync +Vsync
EndSection
```

b. Update **Device** and **BusID** sections by adding the highlighted lines, add as many of these Devices as there are GPU's. This example shows only 2.

```
Section "Device"
Identifier      "Device0"
Driver          "nvidia"
VendorName      "NVIDIA Corporation"
BoardName       "GRID K520"
BusID           "PCI:6:0:0"
Option          "ConnectedMonitor" "CRT-0"
Option          "UseEDID" "FALSE"
EndSection
```

```
Section "Device"
Identifier      "Device1"
Driver          "nvidia"
VendorName      "NVIDIA Corporation"
BoardName       "GRID K520"
BusID           "PCI:7:0:0"
Option          "ConnectedMonitor" "CRT-0"
Option          "UseEDID" "FALSE"
EndSection
```

c. Update **Screen** Section by editing the highlighted lines (this is for 2 GPUs):

```
Identifier      "Screen0"
Device          "Device0"
Monitor         "Headless"
DefaultDepth    24
SubSection      "Display"
    Depth       24
EndSubSection
```

```
Identifier      "Screen1"
Device          "Device1"
Monitor         "Headless"
DefaultDepth    24
SubSection      "Display"
    Depth       24
EndSubSection
```

d. Now, save the file.

6. Start the X Server

a. You can do this by using the command – “startx &”

Connecting to a display

1. Run the command
 - a. `export DISPLAY=:0.0`
 - b. This will set current display to 0

Connecting VNC

1. Now, when everything is done, you need to connect through VNC.
2. Run the following command:
 - a. `x11vnc -display :0`
 - b. This will connect x11vnc to the display 0
3. Note down the address given by x11vnc
 - a. Usually it is shown as “HOSTNAME: <somenum>”
4. As you have the IP address of Linux machine, you may connect to the machine by pointing your VNC Viewer to “IP ADDRESS:<somenum>”
5. First step is to open NVIDIA X-Server settings, and verify it does not show any error.
6. Open a terminal, and run `glxinfo`. `glx vendor string` should show “NVIDIA Corporation” two times.

