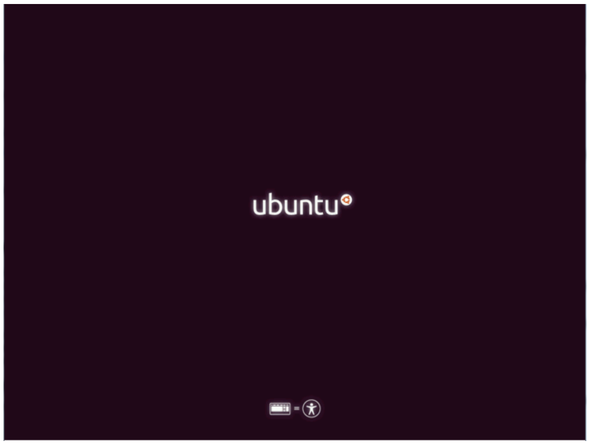
Configuring Baremetal on Ubuntu 12.04

OS Installation:

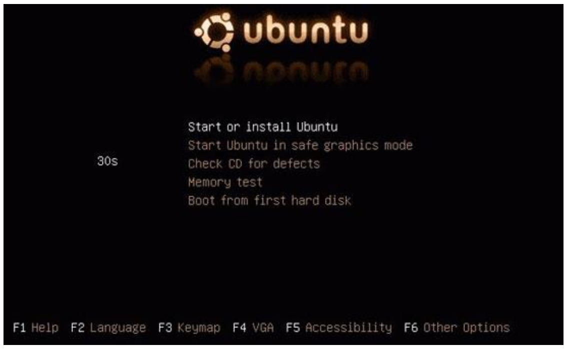
1. Prepare OS installation installer on CD / USB and start the installation. On some systems, to boot to USB, hit F12 during the SBIOS bootup screen, if the menu doesn't come up.



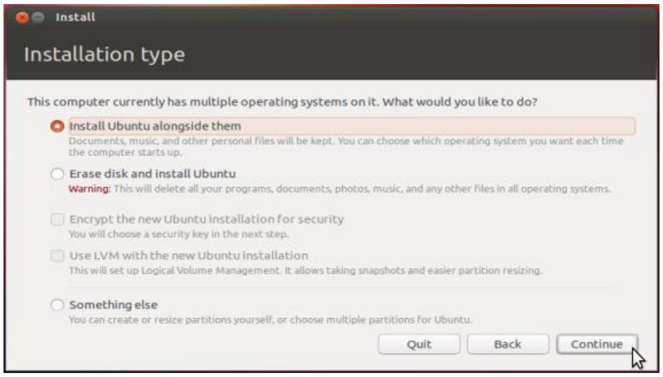
*Note: (**For Ubuntu Desktop only**) - When the installer boots up, you see a graphic (keyboard/user) on the right . Hit F6 to bypass the auto-detection of hardware. This step is not needed for **Ubuntu Server**.*

2. *Another screen comes up with menu to choose the language. Hit ESC to quit and you will see the choices on the right.*

*Note: (**Disabling Nouveau**) – This must be disabled; otherwise there could be installation problems. Hit F6 to get to **Other Options** menu and make sure choose **"acpi off"** and **"nomodeset".** Make sure there is a checkbox, and then hit enter to start Installing Ubuntu. Alternatively you can modify the command line settings and add **"nomodeset"** , which effectively disables the nouveau driver*



3. At the ***Installation Type*** screen, choose "Install Ubuntu alongside them" for a multi-boot HDD. Choose "Erase disk and Install Ubuntu" for single OS HDD.



Install

Installation type

This computer currently has multiple operating systems on it. What would you like to do?

● Install Ubuntu alongside them
Documents, music, and other personal files will be kept. You can choose which operating system you want each time the computer starts up.

○ Erase disk and install Ubuntu
Warning: This will delete all your programs, documents, photos, music, and any other files in all operating systems.

□ Encrypt the new Ubuntu installation for security
You will choose a security key in the next step.

□ Use LVM with the new Ubuntu installation
This will set up Logical Volume Management. It allows taking snapshots and easier partition resizing.

○ Something else
You can create or resize partitions yourself, or choose multiple partitions for Ubuntu.

Connecting remotely via SSH to Linux Server:

- You may want to make sure that the OpenSSH server is installed.

```
ubuntu@ip-10-236-190-141:~$ sudo apt-get install openssh-server openssh-client
ubuntu@ip-10-236-190-141:~$ sudo service ssh restart
```

- For Windows, use Putty to SSH into the server, add the IP address in Host Name.
- 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

For Ubuntu Desktop, 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.

a. Ubuntu 12.04 (Common to Ubuntu Server and Desktop)

```
ubuntu@ip-10-236-190-141:~$ sudo apt-get update
ubuntu@ip-10-236-190-141:~$ sudo apt-get install unzip gcc g++ make pkg-config xserver-xorg build-essential libglew1.6-dev freeglut3-dev libx11-dev libxmu-dev libxi-dev libXxf86vm-dev ia32-libs mesa-utils libgl1-mesa-glx libglu1-mesa libglu1-mesa-dev gcc-multilib ubuntu-desktop x11vnc
```

b. Ubuntu 12.04 (Desktop Only)

```
ubuntu@ip-10-236-190-141:~$ sudo apt-get install kernel kernel-devel
```

c. *Ubuntu 12.04 (Server Only)*

```
ubuntu@ip-10-236-190-141:~$ sudo apt-get install ubuntu-desktop
```

d. *Common to both Server and Desktop*

```
# First create symbolic links for x86_64
ec2-user@ip-10-236-190-141:~$ cd /usr/lib/x86_64-linux-gnu
ec2-user@ip-10-236-190-141:~$ sudo ln -s libGLEW.so.1.6 libGLEW.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libGLEWmx.so.1.6 libGLEWmx.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libXi.so.6 libXi.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libX11.so.6 libX11.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libXrandr.so.2 libXrandr.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libXxf86vm.so.1 libXxf86vm.so
ec2-user@ip-10-236-190-141:~$ sudo ln -s libstdc++.so.5 libstdc++.so

# Repeat to create symbolic links for i386
ec2-user@ip-10-236-190-141:~$ cd /usr/lib/i386-linux-gnu
```

```
#sudo apt-get remove nvidia*
#sudo service lightdm stop
#chmod +x NVIDIA*
#sudo sh ./NVIDIA*.run
#reboot
```

Driver Installation:

1. Kill X-Server
   a. Do this through SSH
   b. `init 3`
   c. `pkill X`
2. Change the mode of NVIDIA installer to execute
   a. `chmod +x` **<NVIDIA Installer Name>.run**
3. Install the driver now
   a. **Sh ./<NVIDIA Installer Name>.run**

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 ftp://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 "**`pkill 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: <somenumber>”
4. As you have the IP address of Linux machine, you may connect to the machine by pointing your VNC Viewer to “IP ADDRESS:<somenumber>”
5. First step is to open NVIDIA X-Server settings, and verify it does not show any error.