# 

[1](#_Toc370366516)

[KeyStone II Workshop Ubuntu Server 6](#_Toc370366517)

[Introduction 6](#_Toc370366518)

[Setup the Network 9](#_Toc370366520)

[Operating system 9](#_Toc370366521)

[64 BIT Server 9](#_Toc370366522)

[Users 10](#_Toc370366523)

[user/local directory 12](#_Toc370366524)

[Building users’ directories 12](#_Toc370366525)

[Utilities 13](#_Toc370366526)

[Global downloads 24](#_Toc370366527)

[Networking 27](#_Toc370366528)

[DHCP Installation 28](#_Toc370366529)

[Student home directory 31](#_Toc370366530)

[Student laptop 31](#_Toc370366531)

# KeyStone II Workshop Ubuntu Server

## Introduction

With the introduction of KeyStone II, TI multicore offering includes ARM CorePac with one, two or four A15 processors. TI offers an EVM card for its customers to speed up development.

Booting KeyStone II EVM is done using U-Boot and requires tftp server available on the same subnet as the EVM. In addition, the toolchain that are used do not run on the target EVM board, but rather on a Linux server. TI uses Ubuntu as the standard Linux operating system.

In KeyStone II workshop we expect to have 10 KeyStone II EVM and about 20 students. Part of the workshop is to develop new code that runs on the A15 Linux system. Since the tool chain runs of Linux, if we want all the students to take part in developing and build application, each student must have access to an Ubuntu machine on the same subnet as the EVM.

TI training equipment includes a set of Windows laptops. To provide Ubuntu access to each student, either Virtual machine must be installed on the Laptops with Ubuntu operating system, or a dedicated Ubuntu server must be available.

While most of the training sessions happen in TI locations, customers training may take place in customer locations. We need to advice a strategy for both cases.

The virtual machine model is strongly discouraged by TI IT team. It presents multiple IP addresses on the same device.

Updating the release and the development tools requires network access with the right proxy settings. Obviously this must be done using the guess network in TI offices (and a similar customers network if the training is in customer site) so that customers cannot have access to internal TI network.

The next page describes the suggested local network organization:

## Server Architecture

### Setup the Network

1. Wireless Network

The wireless network is connected to the external Web. In my case, it is connected to TI network. It is used during installation to download everything. If there is no wireless network, then start by connecting the wire network to the external world, and only after the completion of the installation connect the wire network to the local network 192.168.0.XXX

The wireless network appears as eth2. The interfaces file (/etc/network/interfaces) stay the default. Configure the interfaces using the GUI (setup -> network -> wireless and look at options to set the network, the encryption and so on)

1. Wired network

During the operation, the wire network eth0 is connected to a local switch that supports all the users. The IP address of the server is set to 192.168.0.100. The setting is done using the GUI as explained above. Note, if external network is not available for wireless, the wired network is connected to the external network and gets its IP address from a DHCP server on the network.

Part of the setting is loading a dhcp server that provides IP addresses to all the clients on the local network. Instructions how to install and define the DHCP server are given later in this document. The IP address range is between 192.168.0.10 to 192.168.0.90 – 80 addresses all together.

Note that regardless of the way the server is connected to the external world (wireless or wire), the proxy settings must be changed based on the instructions of the local IT person. In this document the proxy settings are for internal TI network.

### Operating system

TI version of Ubuntu 12.04 with enabled local user is the server operating system. This version is used by TI team to develop the KeyStone Linux code. It is very important to enable super user authority to the local user.

### 64 BIT Server

Some of the downloads that are installed were made for 32-bit machines. In order to run them on 64-bit machines (like the one that we have in North America), one has to install an adaption / compatibility library ia32-libs.

Information on installing ia32-libs is available for example in <http://askubuntu.com/questions/143904/how-to-install-ia32-libs-on-12-04-lts-32-bit>

The following instructions are used to install ia32-lins:

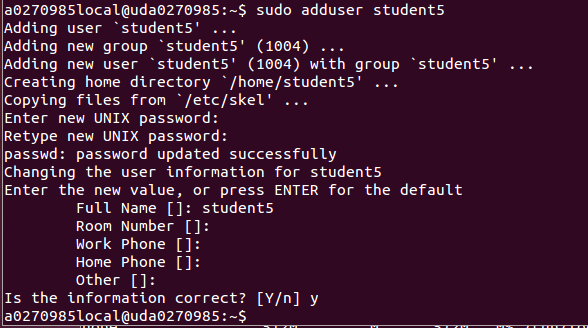
* sudo apt-get install ia32-libs

### Users

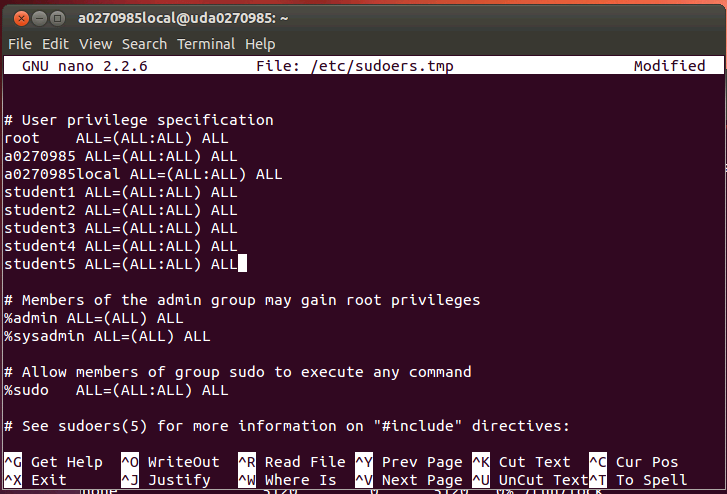
In addition to the root local user with a special password, 12 more users will be defined. The super user for the TI North America system is a0270985local. The password will be given before installing the machine. The following table describes the user names and the password of the other 11 users

|  |  |
| --- | --- |
| User Name | Password |
| student1 | Ws1 |
| student2 | Ws2 |
| Student3 | Ws3 |
| Student4 | Ws4 |
| Student5 | Ws5 |
| Student6 | Ws6 |
| Student7 | Ws7 |
| Student8 | Ws8 |
| Student9 | Ws9 |
| Student10 | Ws10 |
| Student11 | Ws11 |

Instructions how to add a user are given for example in <https://www.digitalocean.com/community/articles/how-to-add-and-delete-users-on-ubuntu-12-04-and-centos-6> . The following screen shots shows how to define a new user



Next we give the new user privileges like root. We do it by adding the new student name to the sudo file using the instruction sudo /usr/sbin/visudo as can be seen in the next screen:



### /user/local directory

/usr/local directory contains all the code and scripts that are used by all the users. I will supply a USB drive with the files that must be loaded into this directory. Two scripts are used by all the users. These scripts should be modified if the external network IS NOT internal TI network. Instructions how to modify these scripts (basically the proxy definitions) will be given by the local IT.

### Building users’ directories

The following process is done to all the user, starting with student1 to student11. The instructions are for student1, expand it to all users is straight forward:

Sudo su - student1

The dash – guarantees that the new user will start with no “left over” configurations from previous sessions.

pwd

ls

The home directory of a new student has only a single file examples.desktop

To finish building the new user account we create two more subdirectories in ~/ (/home/student1)

mkdir applications

mkdir filesys

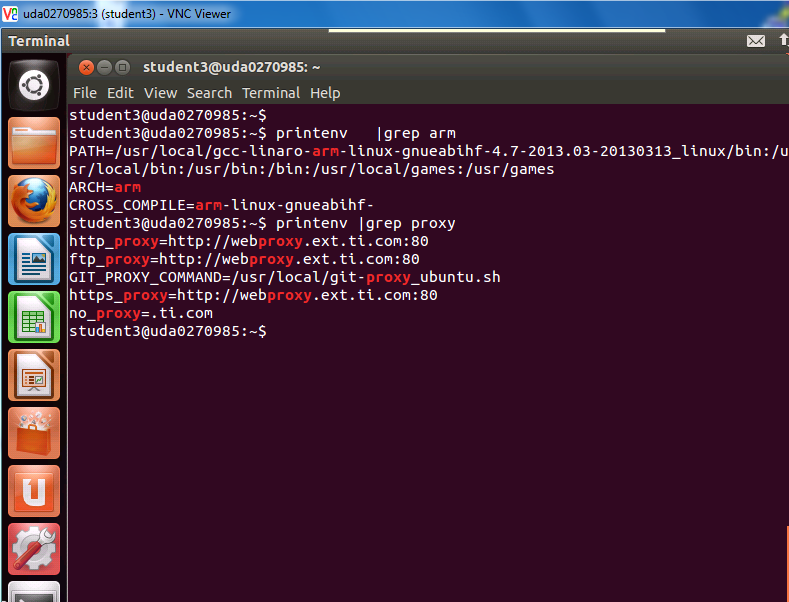
Last we set the environment variables. This is done using a global script. The user must run the script in any new terminal:

source /usr/local/studentStartScript.sh

To verify that the script has worked display some of the new defined path and environment variables. In TI internal network the following proxy will be defined, and the following gcc and arm variables are set

* printenv |grep arm
* printenv |grep proxy

The results should be the same as in the screen shots below (the screen shots were taken from various users. The one below is from student3):



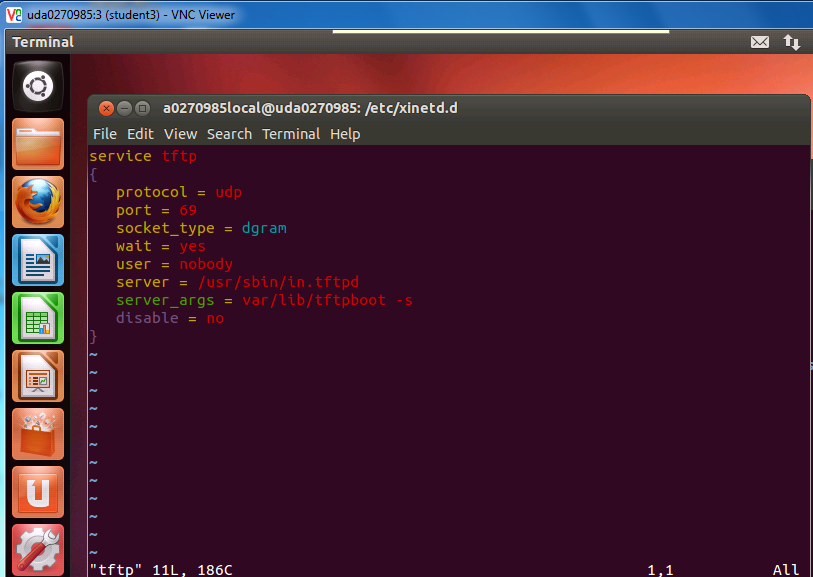
### Utilities

Tftp server, ftp server, nfs server, vnc servers

1. TFTP server

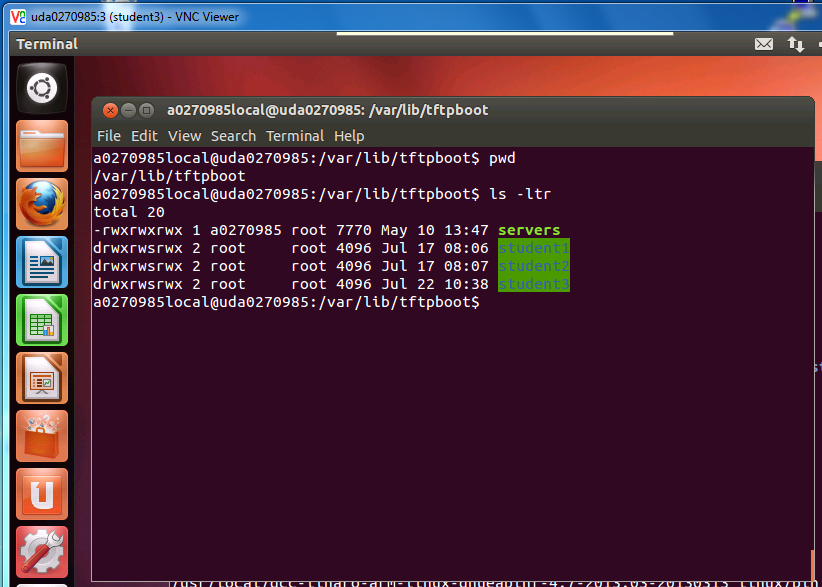
A single TFTP server is needed to support all users. Instruction how to install the TFTP server can be found in <http://icesquare.com/wordpress/how-to-setup-tftp-on-ubuntu/> (even though we use a different Ubuntu release, I followed these instructions).

The install process builds a default tftp configuration file /etc/xinetd.d/tftp. We need to modify this file and define the root directory for tftp downloads. The root directory for all users is /var/lib/tftpboot. The next screen shot display the /etc/xinetd.d/tftp file:



Notice where the server is and the server arguments (no “/” before var)

While /var/lib/tftpboot is the root directory for tftp, we create a subdirectory for each student. The next screen shot shows the /var/lib/tftpboot directory with its sub-directories:

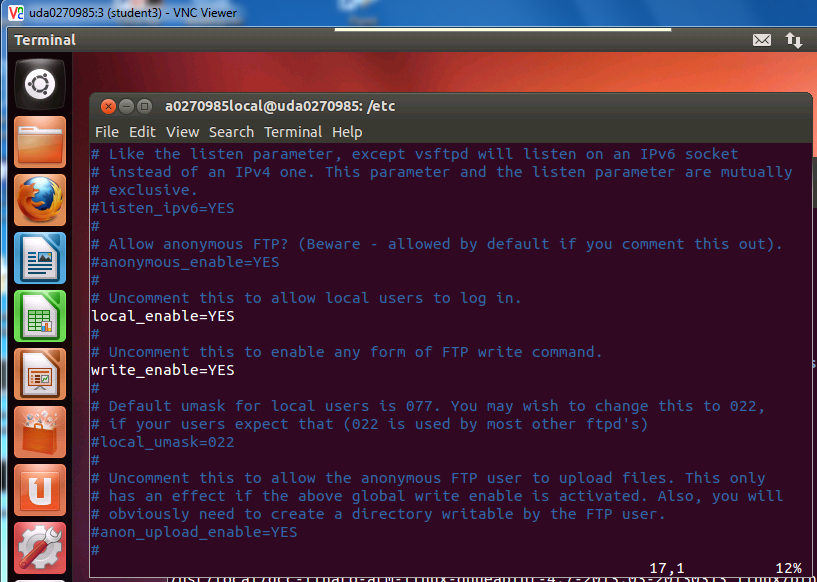


Note – each student will download the boot from the server using its own directory. Instructions how to tell u-boot on each system to get the right boot will be given in the Lab book.

1. ftp server

The VNC server that we will install on the Ubuntu server may not support file transfer. TI enterprise VNC supports file transfer for Window machines and not Linux machines. To move files between the server and individual student Window machines we install ftp server. Instructions how to install the ftp server are given in <https://help.ubuntu.com/10.04/serverguide/ftp-server.html> . A single FTP server supports file transfer for all the users.

After installation of the ftp (sudo apt-get install vsftpd) the file /etc/vsftpd.conf should be modified per the above document. In the following screen shot we show the part of the vsfftd.config file where the local\_enale and write\_enable are enabled and the anonymous enable is commented out.



To start or to reactivate the ftp server the user must use the restart command:

🡪sudo /etc/init.d/vsftpd restart

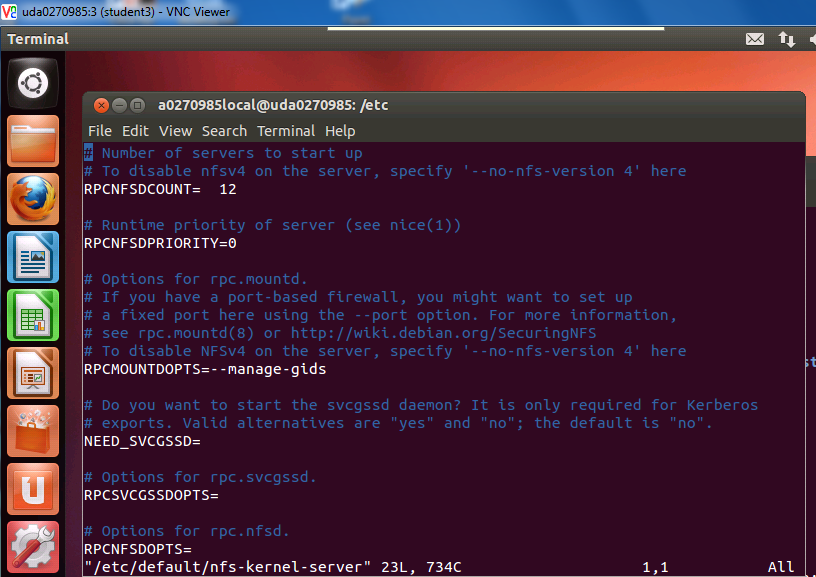
To use the ftp server, each user must login with its own user name and password. The ftp starts at the home directory of the user. The user can switch directories using the cd instructions. Other Linux instructions like ls are also used. Put and get instructions move files between a remote user and the server. Note that access that the user does not have authority is not allowed.

1. nfs server

Similar to the tftp server, the NFS server supports all users, each one from a different sub-directory. Instructions how to install and configure nfs server can be found in <https://help.ubuntu.com/community/SettingUpNFSHowTo>

To install a NFS server and configure it:

* 1. Install the NFS server -> “sudo apt-get install nfs-kernel-server “
  2. Configure the nfs server, open the file /etc/default/nfs-kernel-server The file looks like the following:



Note that when the file is generated, the default value of RPCMOUNTDOPTS is

–manage-gids. The user should change this value and assign a port. Any available port can be used, we use port 13025.

* RPCMOUNTDOPTS = -p 13025

Note that in this setting the firewall is disabled, thus no ufw instruction is needed.

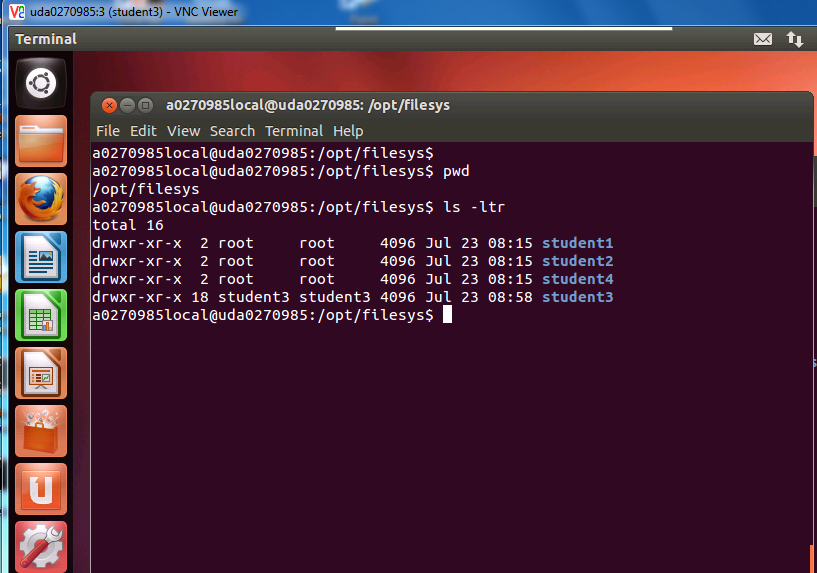
Next the NFS root directory should be created. We use /opt/filesys is the root directory for the NFS

* sudo mkdir /opt/filesys

The root directory should have a sub-directory for each user. Using

* sudo mkdir student1 (and so on)

the sub-directories are created. The next screen shot shows the nfs root directory and the sub-directories (for the first four users):



Note that instructions how to configure the individual EVM to mount on the sub-directory are given in the Lab book.

Last, Add the file system directory to the exports list, open the file /etc/exports and add the following line to it

/opt/filesys \*(rw, subtree\_check, no\_root\_squash,no\_all\_squash,sync)

The file /etc/exports looks like the following:



Start the nfs server

-🡪 sudo /etc/init.d/nfs-kernel-server restart

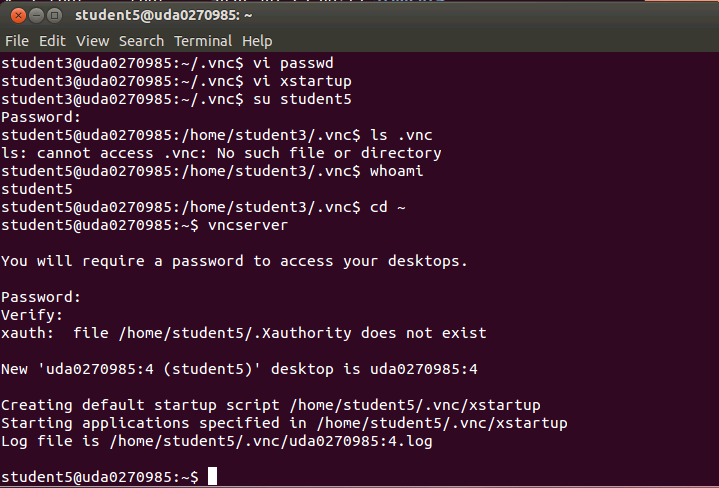
1. vnc servers

Instructions how to install the VNC server can be found in many places, for example: <http://rbgeek.wordpress.com/2012/06/25/how-to-install-vnc-server-on-ubuntu-server-12-04/>

After installing the VNC server, **EACH USER** must define the server from its own home directory and use his own password (6-8 characters). This is done using the following instruction:

vncserver

The server will ask you for a password. We use the same password for all users – vncserve (must by 6 to 8 characters). The following screen shot describes the process:



As you see, the system created a default startup script and started instance number 4. This number is important because when a remote VNC viewer tries to access this account it must specify the instance number in addition to the IP address.

The default script must be replaced by the correct script. The **script xstartup** is provided in the /usr/local directory. To replace the script the following steps are taken:

1. Kill the vnc instance -> vncserver –kill :N where N is the instance number. In the above example N=4
2. Copy the new script to ~/.vnc/xstartup

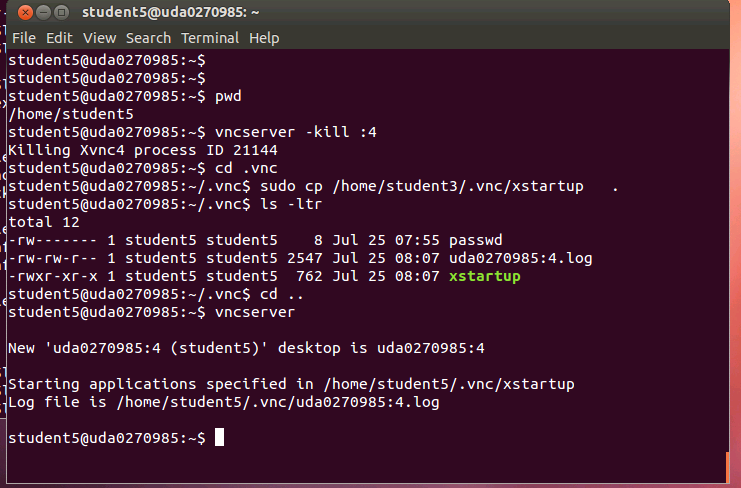
* Sudo cp /usr/local/xstartup ~/.vnc/xstartup

1. Restart the vnc instance

* vncserver

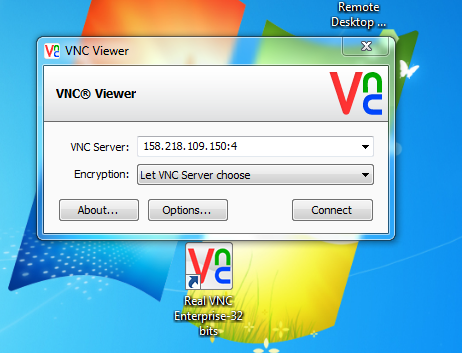
Important Note – before starting step a from above, the user must be “cleaned”. This is done by login into the user account with dash as follows -> “su **–** studentName “

The following screen shot shows the operation on instance number 4 (The xstartup file was copied from student3 and not from /usr/local, it is the same script though):

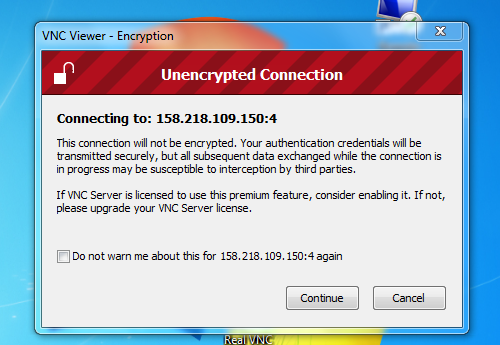


From a remote windows terminal using the VNC viewer, login remotely into the Ubuntu system as student 5 is the following:

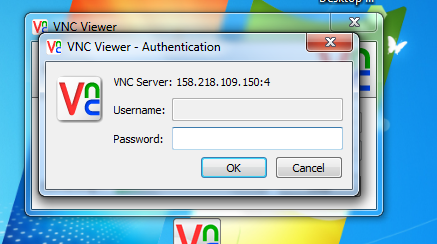
1. Double click on the VNC icon (or start VNC from the start menu)
2. Enter the IP address and the instance number, in the screen shot the ip address is 158.218.109.150 and the instance number is 4. In Lab environment the IP will be something like 192.168.0.100 and the instance number is the student number
3. Let VNC server chose the encryption (no encryption is needed in the lab local network) and click connect



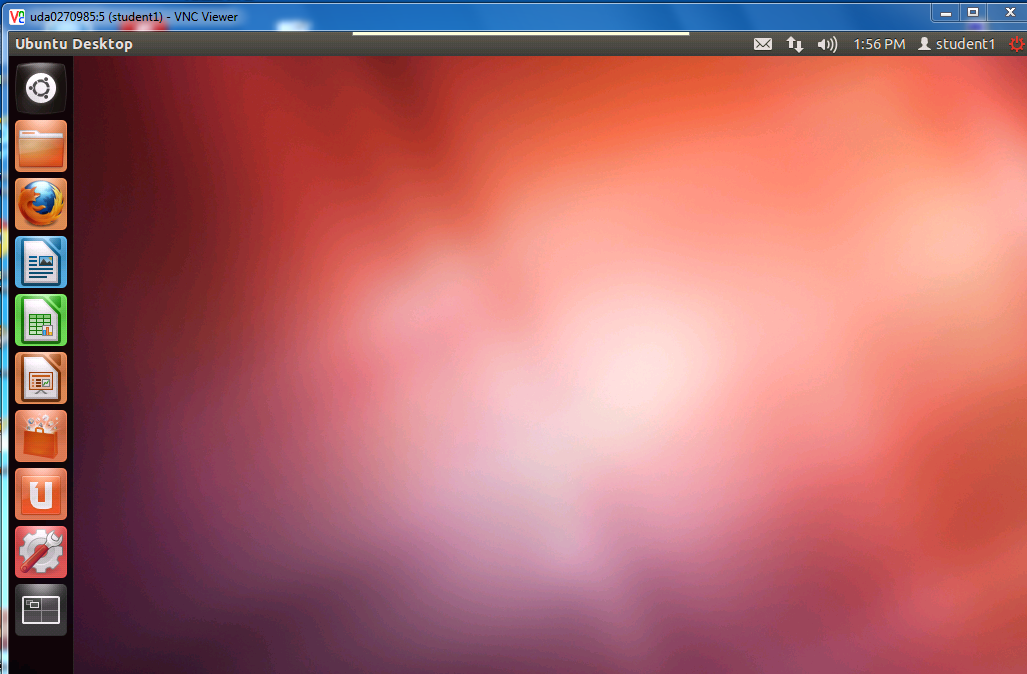
An encryption warning will appear:



Click on continue, a dialogue box will ask for the password:



Enter the password (vncserve) and OK. A window like the following will be opened:



Selecting the dash home (the top square on the left) and left click will open a search tab. Chose Terminal to open a terminal into the user account.

Note, if the VNC is unable to connect, make sure that the Note in the VNC instructions was followed.

Note – if the user error is no authentication, you must start the ypbind server -> sudo restart ypbind

To see what static interfaces are defined for the server look at /etc/network/interfaces

### Global downloads

Some of the software that is needed for executing KeyStone II Lab projects can be shared between multiple users. The software will be installed in a common area and a path to the common area will be defined for each user and each terminal.

1. CCS

As of August 2013 the CCS version that is installed is CCS5\_4. The download location of CCS is <http://processors.wiki.ti.com/index.php/Download_CCS> and it requires the proper licensing and export control approval.

Instructions how to install CCSv5 are here:

* + 1. From the download location in Firefox, download (save as) the CCS compressed file for Linux (ccsv5.4.0.00091\_linux) and the emupack for Linux (ti\_emupck\_keystone2\_setup\_1.0.0.3.bin)
    2. When the download is done, untar the CCS compressed file:

tar xvf CCS5.4.0.00091\_linux.tar.gz

The above instruction generates several files and directories.\

* + 1. Go to the CCS5.4.0.00091\_linux director  
         
       cd CCS5.4.0.00091\_linux
    2. Build the subdirectory to install CCS, /opt/ti/CCS\_5\_4

sudo mkdir /opt/ti/ CCS\_5\_4

* + 1. Add execution permission to the setup file:

sudo chmod +x ccs\_setup\_5.4.0.00091.bin

* + 1. Execute the setup  
         
       sudo ./ ccs\_setup\_5.4.0.00091.bin
    2. The dialogue box asks you where to install. Browse and point to:  
         
       /opt/ti/CCS\_5\_4
    3. To install the emupack, go back to the download directory and install the package

cd .. (take you back to /home/username/Downloads directory)

* + 1. Add execution permission to the setup file

sudo chmod +x ti\_emupack\_keystone2\_setup\_1.0.0.3.bin

* + 1. Execute the setup -> “ sudo ./ ti\_emupack\_keystone2\_setup\_1.0.0.3.bin “
    2. The dialogue box asks you where to install. Browse and point to:  
         
       /opt/ti/CCS\_5\_4/ccsv5/ccs\_base

1. Toolchain

TI uses Linaro toolchain to build executable. The toolchain runs on linux systems.

1. Download the tool chain (August 2013)

<https://launchpad.net/linaro-toolchain-binaries/trunk/2013.03/+download/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313_linux.tar.bz2>

The file will be loaded to the Download directory (/home/user/Downloads). To support multiple users, the tools will be install in a shared directory /usr/local

1. Create the local directory -> “sudo mkdir /usr/local “
2. Move the toolchain tar file into the local directory “ sudo mv gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313\_linux.tar.bz /usr/local/. “
3. Change directory to /usr/local
4. Change the mode of the file “sudo chmod 777 gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313\_linux.tar.bz “
5. untar the file  
     
   sudo tar xjf gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313\_linux.tar.bz2
6. The following three lines should be added to the environment variables for any terminal that is used. The user can add it to the shell file (for example, .bashrc) or have a sh file and source it for each terminal:

export CROSS\_COMPILE=arm-linux-gnueabihf-

export ARCH=arm

PATH=/usr/local/gcc-linaro-arm-linux-gnueabihf-4.7-2013.03-20130313\_linux/bin:$PATH

Note – these lines are part of the studentStartScript.sh script that is in the same directory (/usr/local) and should be run for every terminal.

1. Each user will have his own version of the release taken from the git configuration management. To support git for each user, a new script should be added to the /usr/local directory. The git-proxy-ubuntu.sh script has the following lines:

#!/bin/bash

exec /usr/bin/corkscrew webproxy.ext.ti.com 80 $\*

chmod +x git-proxy-ubuntu.sh

export GIT\_PROXY\_COMMAND=/usr/local/git-proxy-ubuntu.sh

Note that if the session IS NOT in TI office, or the network IS NOT TI network, the two scripts in the /usr/local directory should be changed accordingly

1. MCSDK

A single copy of the latest KeyStone II will be installed in /opt/ti/MCSDK\_3\_XX. Instructions how to load MCSDK are given in <http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Downloads> and for version 11 the download page is <http://software-dl.ti.com/sdoemb/sdoemb_public_sw/mcsdk/latest/index_FDS.html> . The downloads are at the bottom of the page, and MCSDK for Linux as well as the emupack for Linux will be installed.

As was mentions, the installation page is /opt/ti/MCSDK\_3\_XX where XX is the release number.

### Networking

1. Local Network

The Ubuntu server should have access to two networks. The local network 192.168.0.XXX will support all the EVM and the Laptops in the session. An external network, either TI (for internal training) or guess or other network is used to provide connectivity to the git server on the internet. The local network should support at least 12 wired connections for the EVM and 20 wired or wireless connections for students’ Laptops. The local network has a DHCP server on a router or installed on the Ubuntu server.

Note, each EVM is a station, the number of stations is limited to 11.

To restart network do

* sudo restart networking

1. Global network

To support getting individual git repository files the students should have access to the Web. If the training is in TI office and ALL participates are TI employees, the server can be connected to TI network with the local host credentials and the proxy as described in the file studentStartScript.sh. For customer training in TI office a guess network pass is required and the proxy must be available from TI IT team.

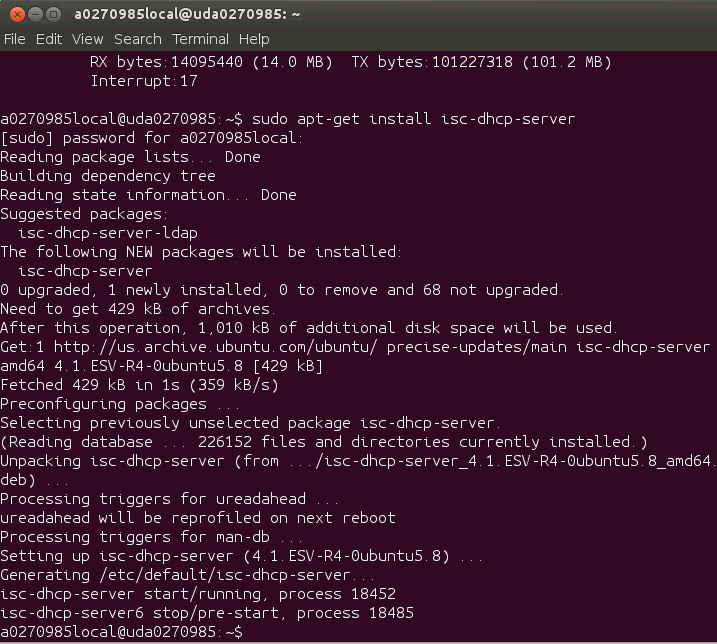
For trainings that are not located in TI office, the local IT person must provide the login credentials and the proxy settings.

## DHCP Installation

Detailed DHCP installation instructions are given in <http://rbgeek.wordpress.com/2012/04/29/how-to-install-the-dhcp-server-on-ubuntu-12-04lts/> the following is a summary of the steps that I did.

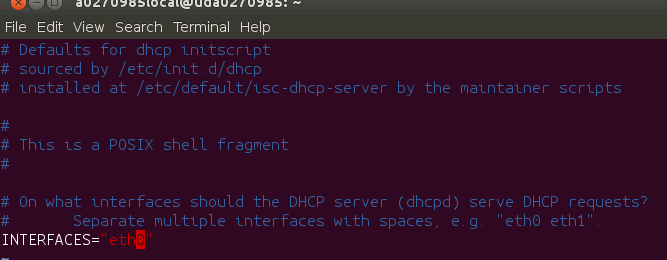
1. Install the server

🡪 sudo apt-get install isc-dhcp-server



1. There are two main dhcp configuration files /etc/default/isc-dhcp-server and /etc/dhcp/dhcpd.conf.
2. Configure /etc/default/isc-dhcp-server. Because the local network 192.168.0.XXX is connected to eth0, the interface will be eth0. If the local network was connected to eth1, the configuration would be different. Do

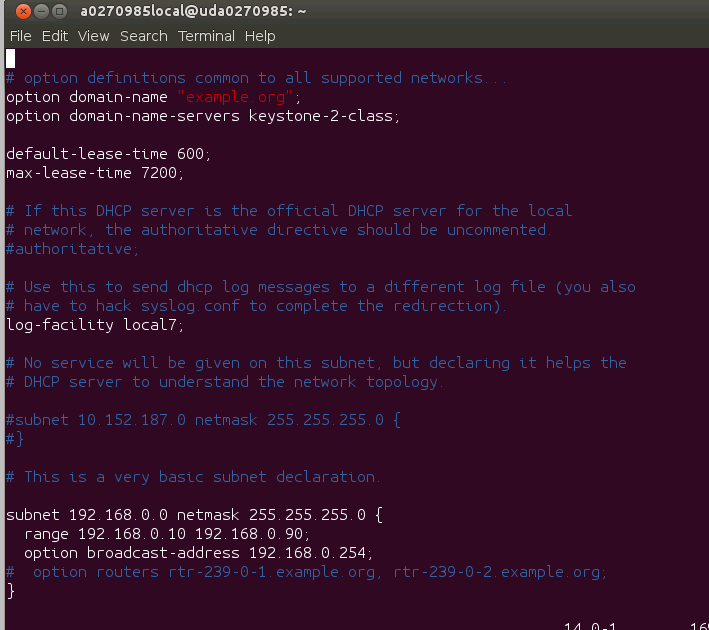
🡪 sudo vi /etc/default/isc-dhcp-server



1. Configure /etc/dhcp/dhcpd.conf file. Do

🡪 sudo vi /etc/dhcp/dhcpd.conf. The following screen shot shows the configuration

* 1. Domain name was defined as keystone-2-class
  2. The range of addresses to give is 192.168.0.10 to 192.168.0.90
  3. Broadcasting address is 192.168.0.254



1. Restart the dhcp server

🡪sudo service isc-dhcp-server restart

1. Checking if the server works correctly

🡪sudo netstat -uap

### Student home directory

The server has a home directory for each EVM. The login name of students is student where N is 1, 2, 3, …11. The password of student is wsN where N is 1, 2, 3, …11. The vnc password is vncserve. The following directories are dedicated for each student:

1. /home/studentName
2. /opt/filesys/studentName
3. /var/lib/tftpboot/studentName

### Student laptop

1. Terminal

It is assumed that each student has a Windows Laptop with VNC viewer installed on it. In addition, each student should have CCS pre-installed (CCSv5.4 or newer) and a GA version of MCSDK3.