

Lab 01: Workspace Setup

Welcome to EE 250L - Distributed Systems for the Internet of Things!

Piazza	2
Course Websites	2
Linux (Ubuntu) Virtual Machine (VM)	2
Identify Architecture	2
Install Virtualization Software, Create Ubuntu VM	3
ARM	3
Intel/AMD	3
Learning the Linux Command Line	3
Updating and Installing Prerequisite Programs	3
Optional: Customize and Install Preferred Software	3
Learning Git and Github	4
Git Tutorials	4
Text Editors	5
Popular Text Editors	5
Flashing Raspberry Pis (rpi)	5
Connect to Your RPi via Secure SHell (SSH)	5
Improving Your Python Skills	6
FAQ	6
Rubric	7

Piazza

A Piazza signup invite has been sent to the entire class via email. If you did not receive the link, please message any instructor or TA or try to enroll yourself [here](https://piazza.com/class/m5lfu7642sw6n7)


<https://piazza.com/class/m5lfu7642sw6n7>

Please use Piazza as an open forum and the main form of communication to the instructor(s), teaching assistant(s), and mentor(s). We will post critical announcements regarding the course on Piazza, so we expect students to check Piazza regularly. Our goal is to always foster a stackoverflow-like flavor to Piazza. Please minimize the use of posts *visible to only instructors*. Ask for help or tips in public posts, and please contribute back. Don't be afraid to contribute even if you are not 100% sure about your answer. Giving it a try is essential to the learning process!

Course Websites

Google Drive and Google Docs will be the main hosting service to host all lab related materials throughout the semester. Click the following link to gain (view only) access to the official student resources folder (you must be logged into your USC account):

Class Google Drive: [Student Resources](#)

Tip: click on  up top to add the folder to your personal Drive (you must use your @usc.edu accounts) for quick access.

For coding assignments, students will submit their code to *Vocareum*. To do this, please sign into your Vocareum.com account.

Linux (Ubuntu) Virtual Machine (VM)

To keep the semester smooth, **we will require everyone to download and run an Ubuntu virtual machine**, even if your laptop is running Linux natively (we will exercise port forwarding into VMs). While you can technically accomplish everything in this course with macOS, production systems in the field almost always use Linux, so this hard requirement will provide you good exposure. In fact, we will be working with Raspberry Pis running Raspbian (a Debian-based Linux distribution, just like Ubuntu) throughout this course.

Identify Architecture

To avoid running into incompatibility issues, it is important for you all to understand the type of processor your computer runs on. The main difference we are concerned with is whether your laptop has an ARM architecture or Intel/AMD. The newer M-series Macs will be ARM, and the

rest will be Intel/AMD. If you don't have a Mac you are almost certainly Intel/AMD. If you still can't tell, ask a TA/CP for assistance before moving on.

Install Virtualization Software, Create Ubuntu VM

The software you use will depend on your processor:

ARM

1. Download and install [UTM](#)
2. Download [Ubuntu Desktop for ARM](#) (Not the AMD64 version!)([Option](#))
3. Follow [this guide](#) to create the VM, using the .iso you just downloaded. Some of the details will differ slightly because we are not using Ubuntu 22.04 server, but the basic steps are the same

Intel/AMD

1. Go to <https://www.virtualbox.org/wiki/Downloads> and download and install VirtualBox **and** the VM VirtualBox Extension Pack.
2. Ubuntu provides [instructions for creating an Ubuntu virtual machine](#).

Learning the Linux Command Line

If you're not familiar with the Linux command line, this resources may be helpful:

- tutorial : http://linuxcommand.org/lc3_learning_the_shell.php
- cheat-sheet: <https://files.fosswire.com/2007/08/fwunixref.pdf>

Updating and Installing Prerequisite Programs

Before your VM is useful, you need to update Ubuntu and install a few tools. To update Ubuntu, run the following commands in the terminal. Do this regularly to keep all your linux packages updated!

```
sudo apt update
sudo apt upgrade
```

Install Git and the UI tools Gitk & Git Gui

```
sudo apt install git
```

Optional: Customize and Install Preferred Software

Since you will be using your VM a lot in the following labs, you may want to spend some time customizing your VM and installing your preferred software. Specifically, you likely want to install

your favorite browser and IDE/text editor. Look up the download page for the software, and find the .deb file corresponding to your processor architecture (Note: not all software will be available for every architecture, make sure you are installing the correct version or you will run into many issues).

Learning Git and Github

Start by making a <https://github.com/> account if you haven't already!

VERY IMPORTANT: Make sure to set your USC email address as the **primary** email for your account and make sure that it is **public** (see the first two sections of this article for help (<https://bytes.usc.edu/github-signup/>)).

During the course, we will distribute code via Github, and you will use a Github repository to backup your code and distribute it to your Raspberry Pi. Git is by far the most used version control system, and its power becomes apparent when you start coding in teams or using it to organize your work. Unfortunately, the benefits are difficult to grasp for a beginner besides the benefit of backing up code. However, we hope to provide exposure to Git's power throughout this course. We suggest formally learning Git through tutorials like the ones below to save you time in this course and beyond. **Not understanding git to the best of your ability will inevitably slow you down in this course!** As you go through the tutorials, you may very well still be confused. This is *absolutely* normal! Nonetheless, it is crucial you run through a tutorial at least once to gain familiarity with the terms.

Git Tutorials

If you have never used git before, we suggest going through whichever learning source fits your style at the following link in addition the material taught in class:

<https://try.github.io/>

Regardless of how you start learning git, you should try to play around with your own toy repository on Github. Lastly, it is highly suggested students complete the **Learn Git Branching** interactive tutorial below to help you visualize what happens as you utilize Git. In particular, we suggest completing "Introduction Sequence" through to "A Mixed Bag" and *all* of the tutorials in the **Remote** tab to better visualize what it means to work with servers. This tool is also great for testing out your git usage flow before actually executing them on your repository.

<https://learngitbranching.js.org/>

Text Editors

If you already know a powerful text editor or IDE (IDEs are overall not recommended), you can skip this section. If you do not, we suggest learning Visual Studio Code from Microsoft.

Popular Text Editors

- [Microsoft's Visual Studio Code](#) (Highly Recommended!)
- [Sublime](#)
- [Atom](#): a free highly-customizable text editor geared towards web developers
- [Micro](#): a modern and intuitive terminal-based text editor
- Vim or GVim (graphical Vim): a terminal text editor
- Emacs: a terminal text editor
- Nano: a terminal text editor for quick tasks for those not well-versed at Vim/Emacs

Flashing Raspberry Pi (rpi)

Recommended method:

STEP 1: Download and install the rpi imager

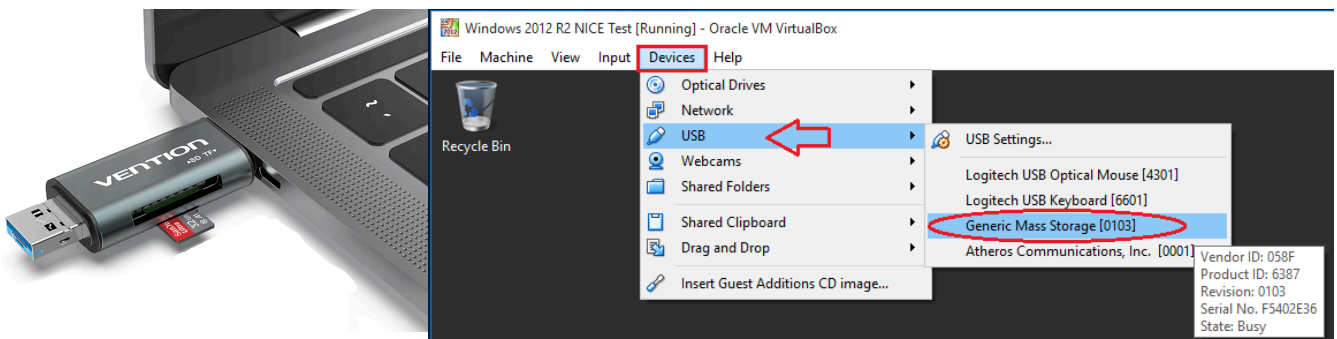
p

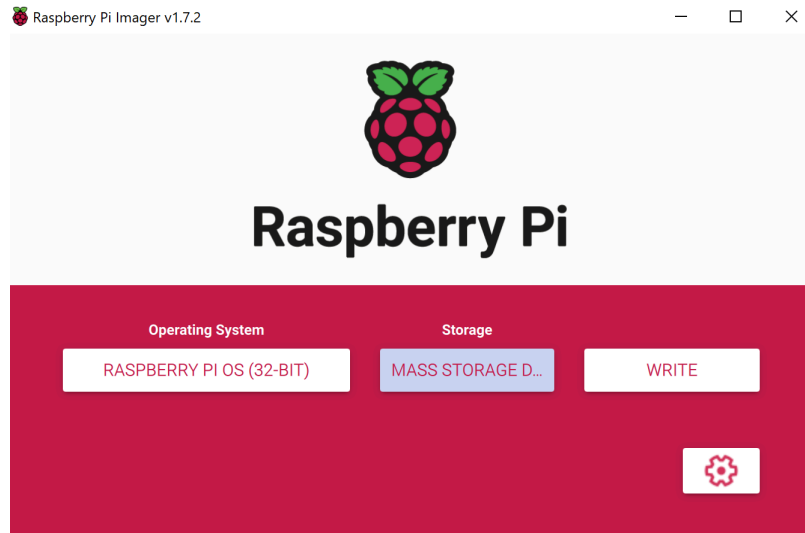
Get it from <https://www.raspberrypi.com/software/>

STEP 2: OS selection and SD storage

Click on the leftmost button and from the list pick the Raspberry Pi OS (32bit) which should be marked as the recommended.

Insert your micro sd card into the computer (using adapter if needed) and click the devices tab on the VM window to forward it into the guest OS. Now in the rpi imager click the middle button and from the options select your sd card.





STEP 3: Advanced options

Click the gears button to access the advanced options menu, here the **hostname** and ssh settings can be configured.

The **hostname** is the name with which your rpi will identify itself within a network. It is suggested to change it to your USC email name (everything before the @usc.edu portion) to avoid having conflicts with multiple hosts with the same name.

Enable SSH with password authentication.

Advanced options X

☒ Set hostname: mwrpi.local

☒ Enable SSH

☒ Use password authentication

☐ Allow public-key authentication only

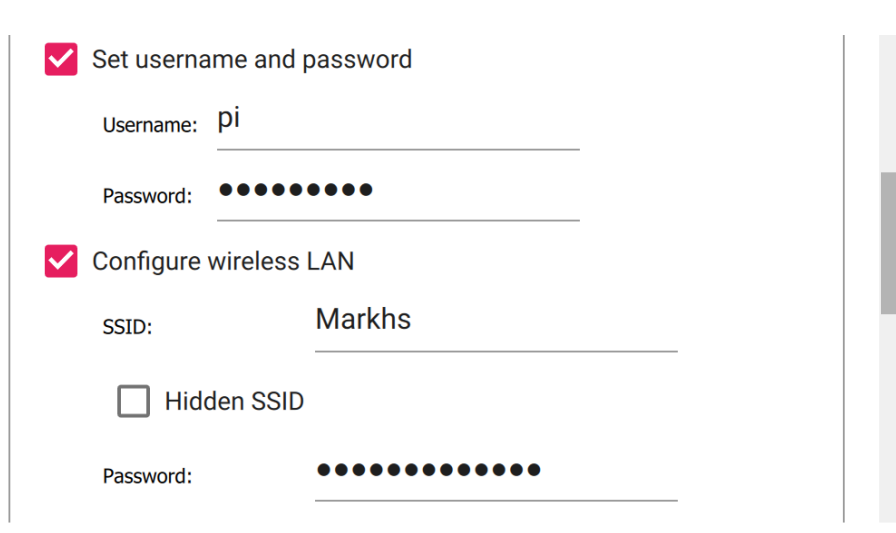
Set authorized_keys for 'pi': /6v6G/Z redekopp@usc.edu

☒ Set username and password

Username: pi

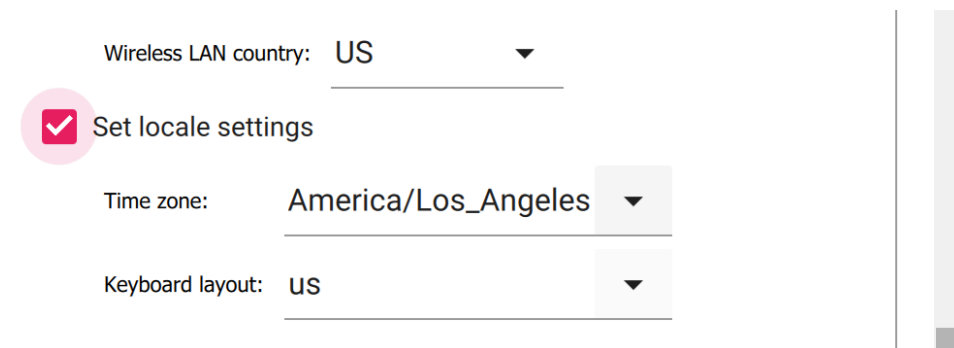
Set a **username** for a session within the rpi with its corresponding password (the default pi name is ok).

In the configuration for the wireless LAN it's **highly recommended** to use your cellphone wifi hotspot credentials.



The screenshot shows the 'Configure wireless LAN' section of the Raspberry Pi configuration tool. It has two checked checkboxes: 'Set username and password' and 'Configure wireless LAN'. Under 'Set username and password', the 'Username' field contains 'pi' and the 'Password' field is masked with dots. Under 'Configure wireless LAN', the 'SSID' field contains 'Markhs', the 'Hidden SSID' checkbox is unchecked, and the 'Password' field is masked with dots. A vertical scrollbar is visible on the right side of the configuration window.

Lastly, configure the regional settings and click save.



The screenshot shows the 'Set locale settings' section of the Raspberry Pi configuration tool. It has a checked checkbox 'Set locale settings'. Above it, the 'Wireless LAN country' dropdown is set to 'US'. Below the checkbox, the 'Time zone' dropdown is set to 'America/Los_Angeles' and the 'Keyboard layout' dropdown is set to 'US'. A vertical scrollbar is visible on the right side of the configuration window.

STEP 4: Click write

Now just click the write button and wait for the flashing and verifying process to finish.

Connect to Your RPi via Secure SHell (SSH)

Insert your newly flashed SD card into your Raspberry Pi, and power it up. For the initial boot, it may take 2-5 minutes before Raspbian sets itself up for the first time. The easiest way to connect to your RPi is to have both your PC and RPi be on the same Local Area Network (LAN).

To access your rpi, you will need to use Secure SHell (SSH). Using a terminal from your VM, SSH into your rpi with your username and the password using a command similar to:

```
ssh    username@hostname           OR
ssh    username@hostname.local      OR
ssh -4 username@hostname           OR
ssh    username@<RPi IP address>
```

#type 'yes' to accept and enter the password when prompted

Demo: Show that you can SSH into your Raspberry Pi in Lab or Lab Hours.

Improving Your Python Skills

We will cover some python basics in class. However, your homework assignment is to learn Python (tackle the topics you feel uncomfortable about!). We will be using Python 3 as much as we can in this course. Some resources with good reputations for beginners include:

<https://www.learnpython.org> (Simple tutorial - Complete the Basics Part)

www.codecademy.com

<https://learnpythonthehardway.org/book/>

<https://developers.google.com/edu/python/>

<https://docs.python-guide.org/> (this one is best used as a reference not tutorial)

As a disclaimer, we have not personally gone through these. Please spend some time trying to write and run a few toy scripts inside your VM to see how to use Python in linux.

Know of an awesome tutorial for beginners? Please share one on Piazza!

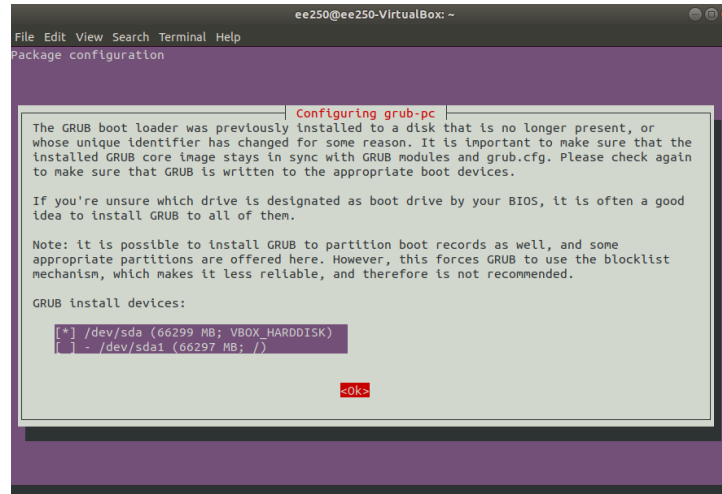
FAQ

Q: How do I resize my VirtualBox VM window?

A: View > Scaled Mode, Click to maximize the window, then change the resolution inside the Ubuntu settings (top right corner > Settings > Display)

Q: The Ubuntu grub menu came up while updating, what do I do?

A: If the grub configuration appears while updating Ubuntu, select only /dev/sda as shown in the screenshot and proceed.



Q: While running `sudo apt update` I got an error saying that my user is not in the sudoers file, how do I fix that?

A: This seems to happen with the unsupervised ubuntu install option, to fix open the terminal and run:

```
~$ pkexec visudo
```

An editor will open find a line that says:

```
root ALL=(ALL) ALL
```

add one with your username below that with the same permissions:

```
username ALL=(ALL) ALL
```

Type `ctrl+x` to close the editor and type `Y` to accept the changes, save the file with the same name prompted.

Rubric

The only deliverable for this lab is to demonstrate to a CP/TA that you can SSH from your VM to your RPi and **get checked off**. Demos can be done in either lab section or office hours.

Note: While we don't explicitly require you to demonstrate your proficiency with git/python/linux in this lab, they will be heavily used in future labs and you will greatly benefit from learning to work comfortably with them ASAP.