

[Wiki »](#)

Setting up a Raspberry Pi at Fermilab

current instructions

The standard 64-bit image is fine. Nice to use the Pi imaging utility to pre-set some settings.

Use raspi-config to switch to X11 and enable screen blanking. Use the GUI config tool to change the keyboard to 104-key, locale to US.

Fermilab network registration: as normal, get serial number from `/proc/cpuinfo`.

Connect to the Fermilab network.

Do an OS update, install basic stuff:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install screen tmux vim-nox emacs-nox krb5-user aptitude saods9 libcfitsio-dev python-is-python3 python3-
```

get the Fermilab krb5.conf:

```
wget https://s3df.slac.stanford.edu/people/meeg/krb5.conf
sudo cp krb5.conf /etc/
```

get recommended dependencies (<https://root.cern/install/dependencies/>), get my prebuilt ROOT:

```
apt-get install binutils cmake dpkg-dev g++ gcc libssl-dev git libx11-dev libxext-dev libxft-dev libxpm-dev python3 li
mkdir Soft; cd Soft
scp meeg@linux-139028:/media/meeg/Seagate/images/root-6.32.02_pi5.tar.gz .
tar -xzf root-6.32.02_pi5.tar.gz
```

in `~/.` bashrc:

```
source $HOME/Soft/root-6.32.02/bin/thisroot.sh
```

Set hostname: edit `/etc/hostname` to match the computer name you used in network registration, e.g. "blueberry"

Edit `/etc/hosts` so it looks like this:

```
127.0.0.1    localhost
127.0.1.1    blueberry.dhcp.fnal.gov blueberry
```

Put the following in `/etc/ssh/sshd_config.d/fnal.conf` (or, find+edit or add in `/etc/ssh/sshd_config`):

```
PubkeyAuthentication no

PasswordAuthentication no

ChallengeResponseAuthentication no

KerberosAuthentication yes
KerberosOrLocalPasswd yes
KerberosTicketCleanup yes

GSSAPIAuthentication yes
GSSAPICleanupCredentials yes
GSSAPIStrictKeyExchange yes

Match Address 192.168.0.0/16
  PubkeyAuthentication yes
  PasswordAuthentication yes
```

The last block re-enables password/pubkey auth on a private LAN. Now enable sshd in the config GUI.

Once you have the password, install the keytab:

```
sudo kadmin -p host/blueberry.dhcp.fnal.gov -w PASSWORD -q "ktadd host/blueberry.dhcp.fnal.gov"
```

old instructions

These instructions are for a Raspberry Pi 4. We usually get the CanaKit Starter MAX kit (case with fan, 64 GB SD card with NOOBS preloaded).

ROOT dependencies:

```
sudo apt-get install science-highenergy-physics-dev libatomic-ops-dev libcfitsio-dev cmake
```

```
mkdir src
cd src
wget https://root.cern/download/root_v6.20.00.source.tar.gz
tar -xzf root_v6.20.00.source.tar.gz
cd
mkdir Soft
cd Soft
mkdir root-6.20.00
cd root-6.20.00
cmake -DCMAKE_SHARED_LINKER_FLAGS='-latomic' ../../src/root-6.20.00/
make -j4
```

Add the following line to ~/.bashrc:

```
source ~/Soft/root-6.20.00/bin/thisroot.sh
```

Add the following to /etc/sysctl.conf (they will not take effect until reboot):

```
net.core.rmem_max=100000000
net.core.rmem_default=100000000
```

IP config (add to /etc/dhcpd.conf, reload with `sudo dhcpcd -n eth1`) for LTA on USB-Ethernet adapter:

```
interface eth1
static ip_address=192.168.133.100/24
```

Kerberos

See [How to kerberize a computer](#).

Configure as a router (not needed to run LTAs, not needed for most people)

<https://www.raspberrypi.com/documentation/computers/configuration.html#setting-up-a-routed-wireless-access-point>

```
apt-get install dnsmasq iptables-persistent
```

in dnsmasq.conf (if you just want the Pi to act as a DNS server, you don't need to touch this):

```
interface=eth1 # Listening interface
dhcp-range=192.168.133.2,192.168.133.20,255.255.255.0,24h
               # Pool of IP addresses served via DHCP
```

Normally, devices that connect to your router will get randomly assigned IP addresses, and you will need to look up that address in order to connect to the device - easiest way is to look at the contents of /var/lib/misc/dnsmasq.leases. If you want any persistently assigned IP addresses (these addresses must be outside the pool you defined with dhcp-range):

```
dhcp-host=11:22:33:44:55:66,192.168.133.61
```

uncomment in /etc/sysctl.d/99-sysctl.conf:

```
net.ipv4.ip_forward=1
```

sysctl.conf won't be applied until reboot, so you should also do `sysctl -w net.ipv4.ip_forward=1`

```
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
netfilter-persistent save
```

You can print iptables rules with `iptables -vL -t nat`.

If you have a ufw firewall enabled:

```
ufw default allow routed
ufw allow in on eth1
ufw route allow in on eth1 out on eth0
```

You could use ufw to also apply the masquerade rule, put this in /etc/ufw/before.rules (see <https://ubuntu.com/server/docs/security-firewall>, <https://gist.github.com/kimus/9315140>):

```
# nat Table rules
*nat
```

```
:POSTROUTING ACCEPT [0:0]

# Forward traffic from eth1 through eth0.
-A POSTROUTING -o eno1 -j MASQUERADE

# don't delete the 'COMMIT' line or these nat table rules won't be processed
COMMIT
```

If you're running the LAN on the built-in Ethernet port and connecting to the Pi over Wi-Fi, substitute (eth1, eth0) with (eth0, wlan0).

For a non-Pi Debian system, Debian 12 (bookworm)

Select the XFCE desktop.

```
adduser sensei sudo
apt install screen tmux vim emacs-nox krb5-user aptitude fwupd
```

For static IP with DNS, install resolvconf and do something like this in /etc/network/interfaces:

```
# The primary network interface
allow-hotplug eno1
iface eno1 inet static
address 131.225.90.24
gateway 131.225.90.1
dns-nameserver 131.225.8.120
dns-nameserver 131.225.17.150
```

```
apt install screen vim-nox emacs-nox git
apt install pkgconf
apt install python-is-python3 saods9 python3-astropy python3-influxdb python3-serial libcfitsio-dev python3-fire
```

ROOT dependencies

there's an official list at <https://root.cern/install/dependencies/> which is probably better than this, but based on <https://www.calel.org/root-bookworm.html>, <https://www.calel.org/root-bullseye.html>, and looking at cmake output for v6.28.04:

```
apt install cmake g++ gcc binutils git python3-dev python3-numpy libx11-dev libxpm-dev libxft-dev libxext-dev libssl-d
```

VNC server

```
apt install tigervnc-standalone-server tigervnc-viewer dbus-x11
```

for headless XFCE sessions:

```
apt install xfce4-session xfce4-panel xfce4-terminal
```

in ~/.vnc/config:

```
localhost
```

in ~/.vnc/xstartup (which should be made executable):

```
#!/bin/sh

unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS
/etc/X11/xinit/xinitrc
startxfce4
```

You might need to kill the local display manager before starting a VNC server: `systemctl stop lightdm`

If you're able to run both a local session and a VNC session, you should disable screen lock.

setting up UPS monitoring with Eaton 5E UPS (this is really only for TAU system):

put this at the end of /etc/nut/ups.conf:

```
[eaton]
driver = usbhid-ups
port = auto
vendorid = 0463
```

put this line in `/etc/modprobe.d/ups-usbhid.conf` (from <https://www.mail-archive.com/nut-upsuser@alioth-lists.debian.net/msg00921.html> and https://networkupstools.org/historic/v2.8.0/docs/FAQ.html#_why_does_my_eaton_5e_usb_ups_on_linux_connect_but_quickly_disconnects_soon):

```
options usbhid quirks=0x0463:0xffff:0x08
```