MAAS Advanced Network Installation and Configuration — Scripted

(The MANIACS Document)

Contents

Purpose

This document describes how to install MAAS on a computer so that you can deploy systems in a test environment as well as install the certification tools and perform certification testing. Consult the Ubuntu Certified Hardware Self-Testing Guide (available from https://certification.canonical.com) for detailed information on running the certification tests themselves.

A computer configured as described here is not intended for general Internet use. Some settings relax security in the interest of ease of use, so you should limit use of the MAAS computer on the Internet at large.

This document begins with information on the required hardware and then moves on to a general description of Ubuntu installation, details on how to install and configure MAAS, and how to test your MAAS installation. Appendixes cover more esoteric topics, including how to add support for CPU architectures other than x86-64 and how to set up advanced network configurations.

Figure 1 illustrates the overall configuration that this document will help you create. This document describes configuration of the MAAS server device in the figure. It presupposes the existence of a local LAN that the MAAS server can use for external connections, as well as the availability of at least one system under test (SUT) for testing at the end of the process. (Note that the Internet connection is required for initial setup, but a properly-configured MAAS server, with local APT repository mirrors, does not need this connection to bring up SUTs.)
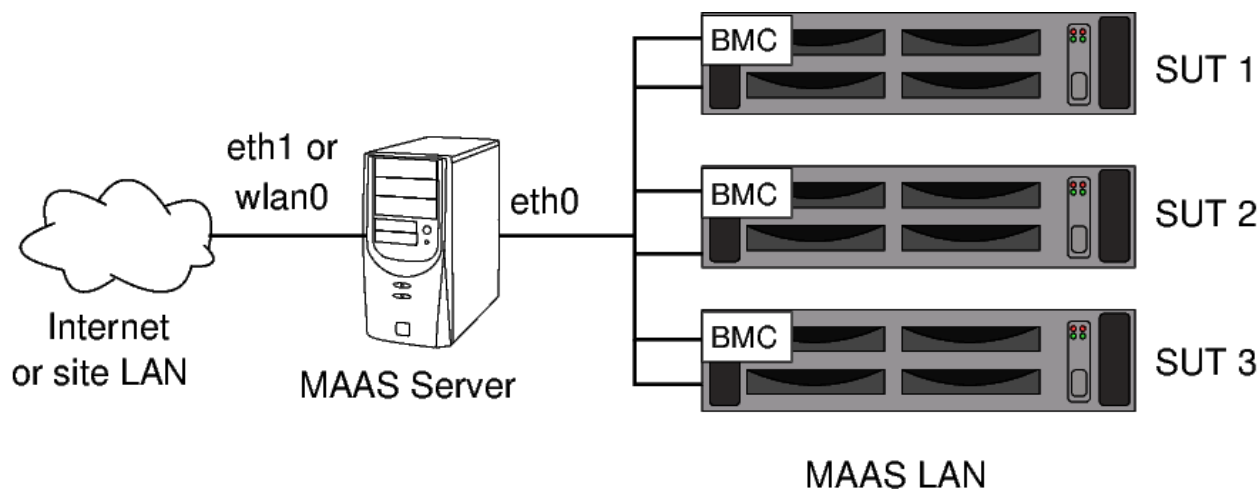
Figure 1: Network structure in which the MAAS server will reside

WARNING: The configuration described in this document leaves several server programs running on the MAAS computer, including a proxy server, a web server, an SSH server, and of course the MAAS server software itself. Thus, it is unwise to expose the MAAS computer directly to the Internet. You should either secure it with strict local firewall rules or place it behind a strong firewall running on a router between it and the Internet.

Prerequisites

Before beginning, you should ensure that your hardware and network meet the following requirements:

- Network
    - The MAAS server should be on its own VLAN or physically segregated network. This is because MAAS will provide DHCP, DNS and PXE services.
    - The network should have the capabilities to match the fastest network device being tested. This will be covered in more detail later.
- MAAS server
    - Ensure that the MAAS server has two network interfaces.
    - At a minmum, the external port should be able to access the Internet while the internal port must be on its own VLAN or physically segregated LAN to avoid conflicts with other network servers providing DHCP, DNS or PXE. Note that external access should be protected as mentioned above in Purpose above.
    - You can install on a virtual machine or container in a more general-purpose computer, but you'll have to pay careful attention to the network and disk settings. Appendix D: Installing MAAS in a LXD Container describes how to set up MAAS in a LXD container.
- System Under Test (SUT) that provides one of the power control types MAAS supports:
    - American Power Conversion (APC) PDU
    - Christmann RECS|Box Power Driver
    - Cisco UCS Manager
    - Digital Loggers, Inc. PDU

- Eaton PDU

- Facebook's Wedge

- HP Moonshot - iLO Chassis Manager

- HP Moonshot - iLO (IPMI)

- IBM Hardware Management Console (HMC) for PowerPC

- IBM Hardware Management Console (HMC) for Z

- IPMI

- Intel AMT

- LXD (virtual systems)

- Microsoft OCS - Chassis Manager

- OpenBMC Power Driver

- OpenStack Nova

- Proxmox

- Redfish

- SeaMicro 15000

- VMWare

- Virsh (virtual systems)

- Webhook

- Switches capable of handling the highest-speed network devices under test.
  - Sufficient Ethernet cables to connect all network ports on the SUTs and the MAAS server, including their BMCs.
  - Be sure cables and switches are capable of handling the fastest network speeds being tested; for instance, if a SUT has a 100 Gbps NIC, you'll need 100 Gbps cables and switches, not 40 Gbps hardware.
  - Please see the Self-Testing Guide for further information on network requirements for certification testing.
- Monitor and keyboard for SUT (helpful, but not strictly required)
- Monitor, keyboard, and optionally a mouse for the MAAS system

- At least 1 TB of disk space with which to mirror the Ubuntu archives, if desired. (An external USB3 hard disk may be used for this, if necessary.)

Note that these hardware requirements are geared toward a typical testing environment. You may need to expand this list in some cases. For instance, if you test network devices of varying speeds, you may need multiple switches and cable types to handle them all.

Installing and Configuring Ubuntu

Once you've assembled the basic hardware for your MAAS server system, you can begin preparing it. The initial steps involve installing Ubuntu and setting up its most basic network settings:

1. Install Ubuntu 20.04 (Focal Fossa) to the MAAS server system.

   - The version of Ubuntu Server 20.04 described here can be obtained from http://releases.ubuntu.com/20.04/. Be sure to download the *live server* version. Any version within the 20.04 series, such as 20.04.3 or 20.04.4, should work. Using the latest point-release available is recommended.

   - Ubuntu may need access to several network sites in order to function properly. These include repositories at `archive.ubuntu.com` (or a regional mirror), Ubuntu's PPA site at `ppa.launchpad.net`, and Ubuntu's key server at `keyserver.ubuntu.com`. (You may instead use local mirrors of the archive and PPA sites.) If your site implements strict outgoing firewall rules, you may need to open access to these sites on ports 80 and/or 443.

   - This guide assumes the use of Ubuntu Server 20.04 and MAAS 3.1. Although other versions of Ubuntu and MAAS may work, some details will differ. Some notable caveats include:

     - **When you boot the installation medium, you should select the "Install Ubuntu Server" option, not any other option.** The procedure in this document involves installing MAAS later.

     - Beginning with Ubuntu 20.04, MAAS is installed via a snap by default, rather than the Debian packages used in earlier versions of Ubuntu; however, this document describes installing MAAS via Debian packages. Use of an older version of Ubuntu is no longer supported, and use of snaps for MAAS is discouraged.

   - On the *Network connections* screen, configure your network ports:

- Configure your *external* network port:
  - If your MAAS server's network devices vary in speed or reliability, use the slower or less reliable device as the external port. This guide assumes this port will be called `eth1`, but in practice it's likely to be something else.

  - Use DHCP or a static IP address, as required by your environment.

  - If you use a static configuration, provide a gateway and DNS server, if possible.

  - In most cases, no explicit configuration of the external port is necessary because the Ubuntu Server installer will have set it up to use DHCP, which is appropriate. You can adjust it if necessary, though.

- Configure your *internal* network port:
  - If your MAAS server's network devices vary in speed or reliability, use the faster or more reliable device as the internal port. This guide assumes this port will be called `eth0`, but in practice it's likely to be something else.

  - This guide assumes use of a static IP address of 172.24.124.1/22 on this port; however, you can use a different network address, if desired or necessary.

  - Using a /22 or wider network is advisable for the internal network, for reasons described in Appendix C: MAAS Network Ranges.

  - If your MAAS server will move from one *external* network to another, be sure to consider all its likely *external* addresses when deciding on its *internal* address and netmask.

  - Avoid the 10.0.3.0/24 address range, because Ubuntu server uses this address range for its LXC container tool.

  - *Do not* set a gateway or DNS server on the *internal* network port.

- If you can't easily differentiate the two ports during installation, you can configure one or both of them after completing the Ubuntu installation. Ubuntu uses NetPlan for network configuration; see https://wiki.ubuntu.com/Netplan/Design and https://netplan.io for details.

- The network configuration screen resembles Figure 2. In this example, `enp0s8` is the internal port and `enp0s3` is the external port.
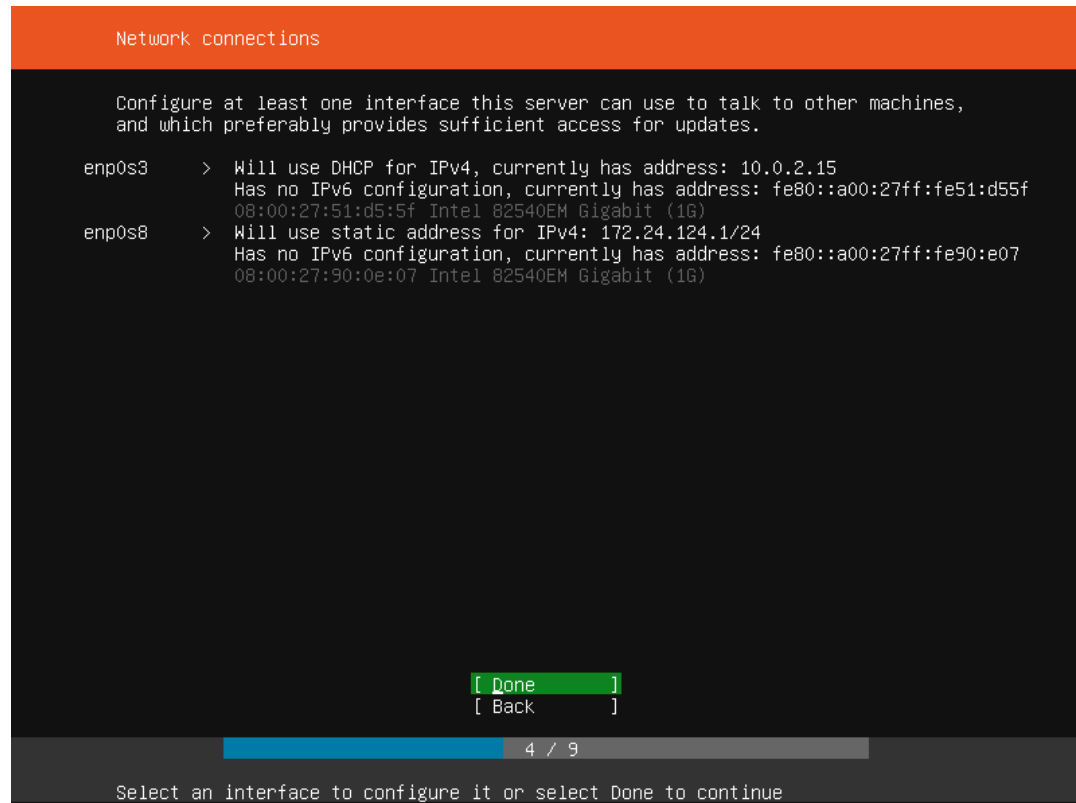

Figure 2: The network can be configured during installation.

- Configure the disk storage and other options as you see fit.

- If you plan to mirror the Ubuntu archives locally, ensure you have enough space in the `/srv` directory to hold your mirrors. As a general rule of thumb, you should set aside about 200 GiB per release. In most cases, a 1 TB disk dedicated to this task works well. If necessary, mount an extra disk at `/srv` to hold your repository mirror. (You can do this after installing Ubuntu, if you like.)

2. When the installation is complete, boot the MAAS computer and log in.

3. Type `ip address` to verify your network configuration. If either network port is not properly configured, edit the configuration file in `/etc/netplan/`. This file may be

called `00-installer-config.yaml`, `01-netcfg.yaml`, or something else; the name depends on the installation method. A typical configuration should look something like this, although likely with different network device names (`eth0` and `eth1` here), possibly different IP addresses, and perhaps with more or fewer options:

```
network:
  version: 2
  ethernets:
    eth0:
      match:
        macaddress: 24:8a:07:a3:18:fc
      addresses:
      - 172.24.124.1/22
      nameservers: {}
      mtu: 9000
    eth1:
      dhcp4: true
```

If your network includes any high-speed network devices (above 10 Gbps), you may need to add `mtu: 9000` to that device's configuration, and possibly tie the definition to a specific MAC address, as shown in the `eth0` definition. Additional information on testing with such devices is in Appendix D of the Ubuntu Certified Hardware Self-Testing Guide (available from https://certification.canonical.com).

If you need to change the network configuration, type `sudo netplan apply` or reboot the computer to apply the changes.

4. Update the software on your system to the latest versions available:

```
$ sudo apt update
$ sudo apt dist-upgrade
```

5. If desired, install X11 and your preferred desktop environment. This will enable you to use the MAAS computer itself to access the MAAS web UI. You can skip this step if your MAAS server will be accessed remotely. If in doubt, don't install X11 and a desktop environment. You can always install it later if you discover it's necessary. In

most cases, you can install X11 and the desktop environment with a single command, such as the following to install Ubuntu 20.04's GNOME:

```
$ sudo apt install vanilla-gnome-desktop
```

6. Reboot the computer. This enables you to begin using your updated kernel (if it was updated) and ensures that your network settings will survive a reboot.

Installing and Configuring MAAS

Installing MAAS on the computer is quite straightforward; you simply use APT. With MAAS installed, you can run the `maniacs-setup` script to configure MAAS for use in an Ubuntu certification environment.

Installing MAAS

Configuring MAAS is described in generic terms at http://maas.ubuntu.com/docs/install.html. The more specific procedure for using MAAS in certification testing is:

1. Enable the MAAS 3.1 PPA:

```
$ sudo apt-add-repository ppa:maas/3.1
```

2. Several scripts and configuration files are available in the `maas-cert-server` package in the hardware certification PPA. You can install the scripts and configuration files as follows:

```
$ sudo apt-add-repository ppa:hardware-certification/public
$ sudo apt install maas-cert-server
```

The `maas-cert-server` package includes a dependency on MAAS, so installing `maas-cert-server` will also install MAAS, as well as all of MAAS's dependencies. Note that the installation process will warn you that MAAS must be set up after installation. **Do not** set it up with the specified command; that will be handled later.

3. Verify that MAAS is installed:

```
$ apt-cache policy maas | grep Installed
```

The output should specify the installed MAAS version, which should be 1:3.1. If this command reports that a very old-looking version, such as 1:0.7, is installed, then chances are something went wrong when adding the MAAS PPA, and the snap version is installed. You can verify this by checking for the snap package:

```
$ snap info maas | grep installed
```

If MAAS is installed via a snap, then it is recommended to:

1. Uninstall the snap (`sudo snap remove maas`)
2. Fix the PPA configuration (`sudo apt-add-repository ppa:maas/3.1`)
3. Re-install MAAS (`sudo apt install maas`).

4. Edit the `/etc/maas-cert-server/config` file to be sure that the variables it contains are correct. Specifically:
   - `INTERNAL_NET` must point to your *internal* network device (`eth0` in the below examples).
   - `EXTERNAL_NET` must point to your *external* network device (`eth1` in the below examples).
   - Do not adjust other values without consulting with the Server Certification Team.
   - Note that there must *not* be spaces surrounding the equal signs (=) in the assignments!

5. Optionally create an `/etc/maas-cert-server/iperf.conf` file to identify your `iperf3` server(s). (If you must install MAAS via a snap, this file should be `/var/snap/maas/current/iperf.conf`.) This file should consist of a single line that contains a comma-delimited list of IP addresses, each identifying a different `iperf3` server. If this file is absent, SUTs will configure themselves to use their network gateways (normally the MAAS server) as the `iperf3` target. If `/etc/maas-cert-server/iperf.conf` is present, though, MAAS will tell SUTs to use the specified system(s) instead. You might use this feature if your `iperf3` server is

not the SUTs' network gateway or if you have multiple `iperf3` servers. The SUTs will attempt to use each `iperf3` target in series until the network test passes or until the list is exhausted. This setting can be overridden on SUTs by editing the `/etc/xdg/canonical-certification.conf` file on the SUT. See Appendix B: Network Testing Options for more on advanced network testing configurations.

Running the Setup Script

This section describes setting up MAAS directly on the MAAS server computer's hardware. If you prefer to run MAAS within a LXD container for added flexibility, consult Appendix D: Installing MAAS in a LXD Container.

The MAAS configuration script is called `maniacs-setup`, and was installed as part of the `maas-cert-server` package. Running this script will set up the MAAS server with reasonable defaults for certification work; however, the script will also ask you a few questions along the way:

```
$ sudo maniacs-setup

************************************************************************
* Identified networks:
*    INTERNAL: 172.24.124.1 on eth0
*    EXTERNAL: 192.168.1.27 on eth1
*
* Is this correct (Y/n)?
```

Be sure your network assignments are correct at this point! If the script complains about a problem, such as an inability to identify an IP address or a default route being present on your internal network, go back and review both your network settings and the contents of your `/etc/maas-cert-server/config` file to identify the cause and correct the problem.

If you approve the settings, the script will display additional messages as it begins to configure the MAAS server. Some of these messages are the output of the programs it calls. For the most part this output can be ignored, but if a problem occurs, be sure to report it in detail, including the script's output.

Note that at all prompts for a "Y/N" response, the default value is capitalized; if you press Enter, that default will be used.

If you've installed MAAS via snaps, the next question acquires a password for a PostgresQL database (with the name *maas*) upon which MAAS relies:

```
**************************************************************************
* We must set up a PostgresQL account (called 'maas') with a password that
* you supply.
*
* Please enter a password for this account:
* Please re-enter the password for verification:
```

MAAS records this password and uses it itself, so ideally you won't need it; but you may need it if you must perform manual database maintenance, so it's best to remember the password.

The script will now ask for a password for the MAAS administrative account, which will have the same name as your default login name. Note that this account is distinct from the PostgresQL account created earlier.

```
**************************************************************************
* A MAAS administrative account with a name of ubuntu is being
* created.
*
* Please enter a password for this account:
* Please re-enter the password for verification:
```

In most cases, you should enable NAT on your MAAS server; however, if official policy at the site where the server will be used forbids the use of NAT, you may opt to leave it disabled:

```
**************************************************************************
* NAT enables this computer to connect the nodes it controls to the Internet
* for direct downloads of package updates and to submit certification results
* to C3.
*
* You can configure this computer to automatically start NAT. The following
* commands can start or stop NAT:
*   sudo systemctl enable certification-nat -- Start NAT on the next reboot
```

```
*  sudo systemctl disable certification-nat -- Stop NAT on the next reboot
*  sudo service certification-nat start -- Start NAT until the next reboot
*  sudo service certification-nat stop -- Stop NAT until the next reboot
*
* Do you want to set up this computer to automatically enable NAT (Y/n)?
```

The `service certification-nat start` and `service certification-nat stop` commands run the `startnat.sh` and `flushnat.sh` scripts, respectively. Both of these scripts come with the `maas-cert-server` package. The `systemctl` commands set up or remove symbolic links in the systemd configuration directories to enable (or not) NAT when the computer boots. You can check whether NAT is running by typing `sudo iptables -L`, which should show a pair of `ACCEPT` rules in the `FORWARD` chain if NAT is enabled and no such rules if it's not running; and by typing `cat /proc/sys/net/ipv4/ip_forward`, which should return `1` if NAT is enabled and `0` if it's not enabled.

If your work site has poor Internet connectivity or forbids outgoing connections, you must create a local mirror of the Ubuntu archives on your MAAS server. These archives will be stored in the `/srv/mirrors/` directory, but creating them takes a long time because of the amount of data to be downloaded — about 200 GiB per release. For comparison, HD video consumes 1-8 GiB per hour — usually on the low end of that range for video streaming services. As should be clear, the result will be significant network demand that will degrade a low-end connection for hours, and possibly exceed your monthly bandwidth allocation. The download will occur in the background, though, so you can continue with MAAS setup as the download proceeds. If you want to defer creating a mirror, you should respond `N` to the following prompt, then re-launch `maniacs-setup` with the `--mirror-archives` (or `-m`) option later. In any event, you make your selection at the following prompt:

```
***********************************************************************
* Mirroring an archive site is necessary if you'll be doing testing while
* disconnected from the Internet, and is desirable if your test site has
* poor Internet connectivity. Performing the mirroring operation takes
* time and disk space, though -- about 150 GiB per release mirrored.
* To defer this task, respond 'N' to the following question.
*
* Do you want to mirror an archive site for local use (y/N)? Y
```

If you opt to mirror the archive, the script will ask you to verify the upstream mirror site:

```
* Identified upstream archive is:
*  http://us.archive.ubuntu.com/ubuntu/
*
* Is this correct (Y/n)? y
```

If you respond `n` to this question, the script asks you to specify another archive site. The script then asks you which Ubuntu releases to mirror:

```
* Do you want to mirror bionic (Y/n)? n
* Do you want to mirror focal (Y/n)? y
* Do you want to mirror impish (Y/n)? n
* Do you want to mirror jammy (Y/n)? y
```

The list of releases changes as new versions become available and as old ones drop out of supported status. When the mirror process is done, you'll be asked if you want to configure the computer to automatically update its mirror every day, by modifying the `/etc/cron.d/apt-mirror` file. If you do not opt for automatic daily updates, you can update your mirror at any time by typing `sudo apt-mirror`.

```
* Set up cron to keep your mirror up-to-date (Y/n)? y
* Cron should update your mirror every morning at 4 AM.
* You can adjust /etc/cron.d/apt-mirror manually, if you like.
```

Note that `maniacs-setup` configures the system to mirror AMD64, i386, and source repositories because all three are required by the default MAAS configuration. If you want to tweak the mirror configuration, you can do so by editing the `/etc/apt/mirror.list` file — but do so *after* finishing with the `maniacs-setup` script, and then type `sudo apt-mirror` to pull in any new directories you've specified. You can also configure the computer to use its own local mirror, if you like:

```
* Adjust this computer to use the local mirror (Y/n)? y
```

The script then gives you the option to retrieve images used for virtualization testing. If your site has good Internet connectivity, you may not need these images; but it's not a bad idea to have them on hand just in case. Although downloading the cloud images isn't

nearly as time-consuming as mirroring the archives, it can take long enough that you may want to defer this action. You can download the cloud images later by launching `maniacs-setup` with the `--download-virtualization-image` (or `-d`) option.

```
************************************************************************
* An Ubuntu cloud image is required for virtualization tests. Having such
* an image on your MAAS server can be convenient, but downloading it can
* take a while (each image is about 250MiB). This process will import cloud
* images for whatever releases and architectures you specify. If you select
* 'Y', logs will be stored at /home/ubuntu/.maas-cert-server/cloudimg-*-dl-*.log;
* monitor them if you suspect download problems.
*
* To defer this task, respond 'N' to the following question.
*
* Do you want to set up a local cloud image mirror for the virtualization
* tests (Y/n)?
```

If you respond Y to this question, the script proceeds to ask you what Ubuntu versions and architectures to download:

```
* Cloud Mirror does not exist. Creating.
* Do you want to get images for bionic release (y/N)? n
* Do you want to get images for focal release (Y/n)? y
* Do you want to get images for impish release (y/N)? n
* Do you want to get images for jammy release (y/N)? n
*
* Do you want to get images for amd64 architecture (Y/n)? y
* Do you want to get images for i386 architecture (y/N)? n
* Do you want to get images for arm64 architecture (y/N)? n
* Do you want to get images for armhf architecture (y/N)? n
* Do you want to get images for ppc64el architecture (y/N)? n
* Do you want to get images for s390x architecture (y/N)? n
* Downloading cloud images.  This may take some time.
*
* Downloading image for focal on amd64 in the background....
```

These downloads proceed in the background, with logs stored in `~/.maas-cert-server`, so you can check there if you suspect problems. To monitor the downloads, use `top` or `ps` to look for instances of `wget`.

You can customize the site that MAAS tells nodes to use for their repositories. If you mirrored a repository, the script points nodes to itself (via its internal IP address); but if you did not mirror a repository, the script should point your nodes to the same site used by the MAAS server itself. In either case, you can press the Enter key to accept the default or enter a new value:

```
************************************************************************
* MAAS tells nodes to look to an Ubuntu repository on the Internet. You
* can customize that site by entering it here, or leave this field blank
* to use the default value of http://172.24.124.1/ubuntu.
*
* Type your repository's URL, or press the Enter key:
```

At this point, the script tells MAAS to begin importing its boot resources — images it uses to enlist, commission, and deploy nodes. This process can take several minutes to over an hour to complete, but it will happen in the background, so it doesn't ask for permission, but it does tell you what it's doing:

```
************************************************************************
* MAAS requires boot resource images to be useful, and they will now be
* imported in the background. This can take a LONG time, but will not
* significantly slow down subsequent configuration steps.
```

If the download of boot resources fails and you want to initiate it manually later, you can use the MAAS web UI or launch `maniacs-setup` with the `--import-boot-resources` (or `-i`) option.

Sometimes this process freezes. Typically, the boot images end up available in MAAS, but the script doesn't move on. If this happens, you can kill the script and re-launch it to finish the installation.

The script configures personal package archives (PPAs), in which the latest certification software is stored. You'll want to configure PPAs for whatever architectures you intend to test:

```
**********************************************************************
* Now we will set up the PPAs necessary for installing the certification
* tools when deploying the SUT.
*
* Do you want to set PPAs for amd64 architecture (Y/n)? Y
* Do you want to set PPAs for i386 architecture (y/N)? n
* Do you want to set PPAs for arm64 architecture (y/N)? n
* Do you want to set PPAs for armhf architecture (y/N)? n
* Do you want to set PPAs for ppc64el architecture (y/N)? n
*
* Adding PPAs for the following architectures: amd64
*
* Hardware Certification Stable PPA
* Firmware Test Suite Stable PPA
* Hardware Certification Development PPA (Disabled by default)
*
* PPA Setup Complete
```

Finally, the script announces it's finished its work:

```
**********************************************************************
* The maniacs-setup script has finished!
```

In addition to setting the options for which it prompts, maniacs-setup adjusts some other details of which you should be aware:
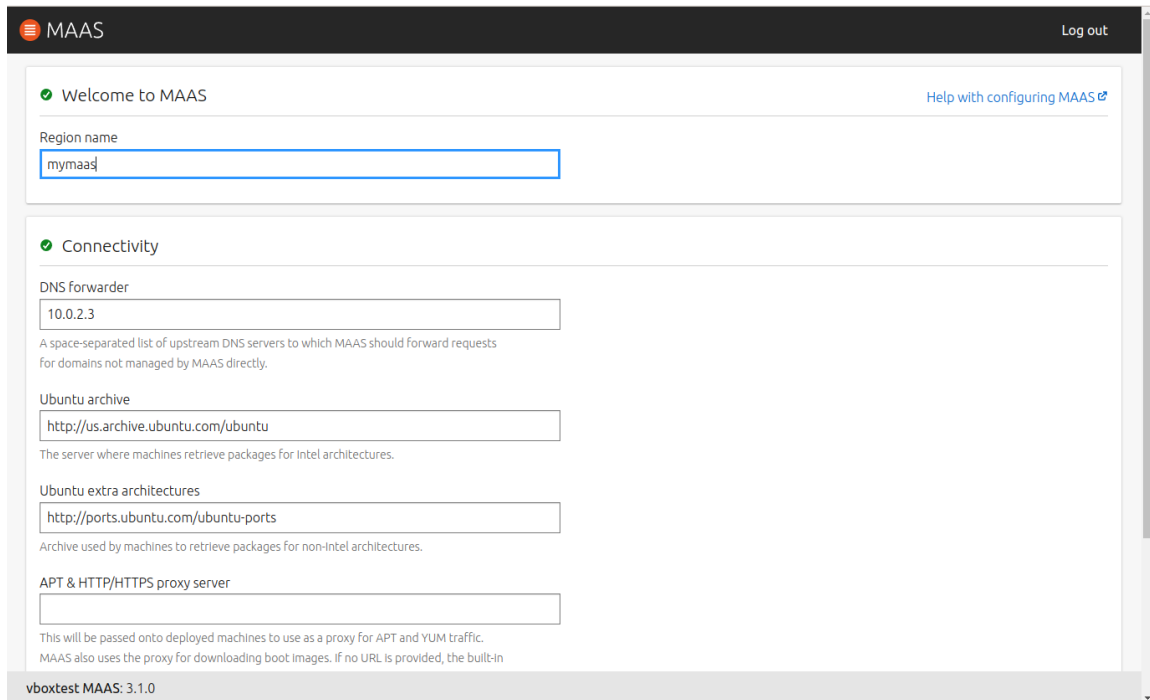
- SSH keys are generated for your user account and added to the MAAS server. These keys enable you to log in to nodes that MAAS deploys from your regular account on the MAAS server.

- Any keys in your ~/.ssh/authorized_keys file on the MAAS server computer are also added to the MAAS setup. Again, this simplifies login.

- The MAAS computer's SSH client configuration is relaxed so that changed host keys do not block outgoing connections. This change is *insecure*, but is a practical necessity because your internal network's nodes will be redeployed regularly. You should keep this setting in mind and minimize your use of this computer to SSH to external sites.

- MAAS is configured to tell nodes to install the Canonical Certification Suite whenever they're deployed. This detail increases deployment time compared to a generic MAAS installation.
- The default storage layout setting is changed from "LVM" to "flat." Some certification tests assume a flat layout.

Checking the MAAS Configuration

At this point, MAAS should be installed and configured; however, it's worth verifying the most important options in the MAAS web UI. You may also want to modify a few settings. To do so, follow these steps:

1. Verify you can access the MAAS web UI:
    - Launch a browser and point it to http://172.24.124.1:5240/MAAS (changing the IP address as necessary).
    - Note that MAAS 2.3 and earlier accepted connections on port 80, but starting with MAAS 2.4, you *must* use port 5240.
    - You should be able to access the server on either its internal or external network address, although at this point, the only computer on the internal network may be the MAAS computer itself.
    - If you provide the computer with a hostname in DNS or /etc/hosts, you should be able to access it via that name, as well.
    - You should see a login prompt.
2. Log in to the web UI using your regular username and the password you gave to the setup script.
3. Once you log in, MAAS presents a screen in which you can set a few options, as shown below. Review these settings, changing any as necessary, and click Save and Continue at the bottom of the page. (If in doubt, leave the settings as-is; you can change them later, if necessary.)
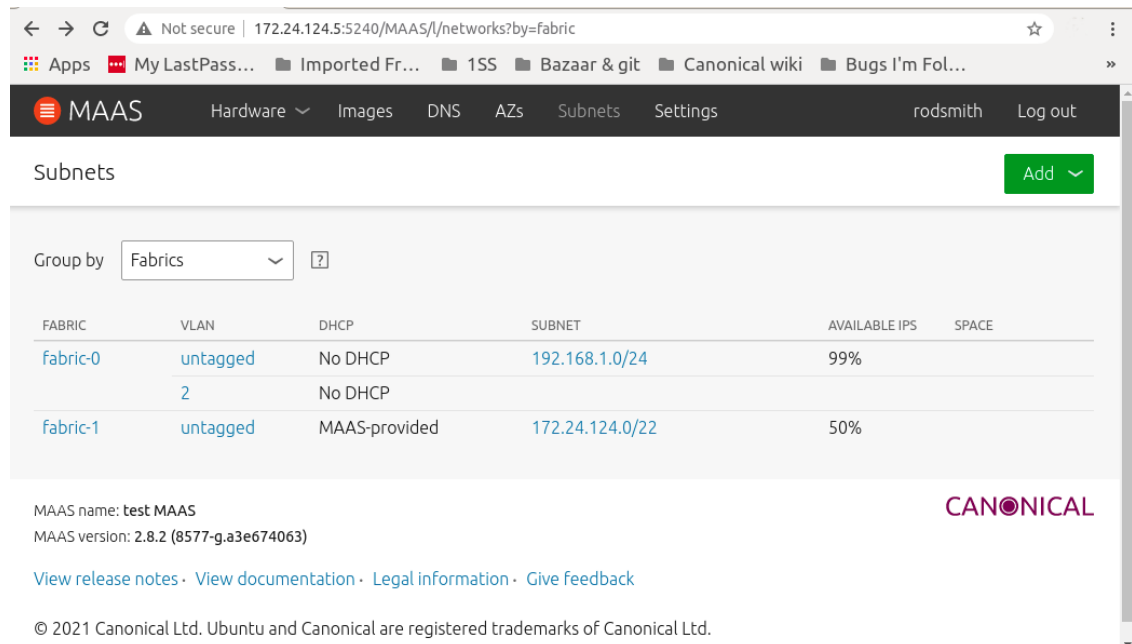
4. MAAS now shows a list of OS images, as shown below. This page will probably show Ubuntu 20.04 for AMD64 already synced or importing. You can select additional Ubuntu releases using the radio buttons, and for each release, pick the architectures you want to import. You probably don't need to do anything with this page right now, and you can come back to it later; however, if you know you must test with an unusual architecture or something other than Ubuntu 20.04, you may want to import additional images immediately. Click Update Selection once you've made your picks. When you're done making changes, scroll down and click Continue.
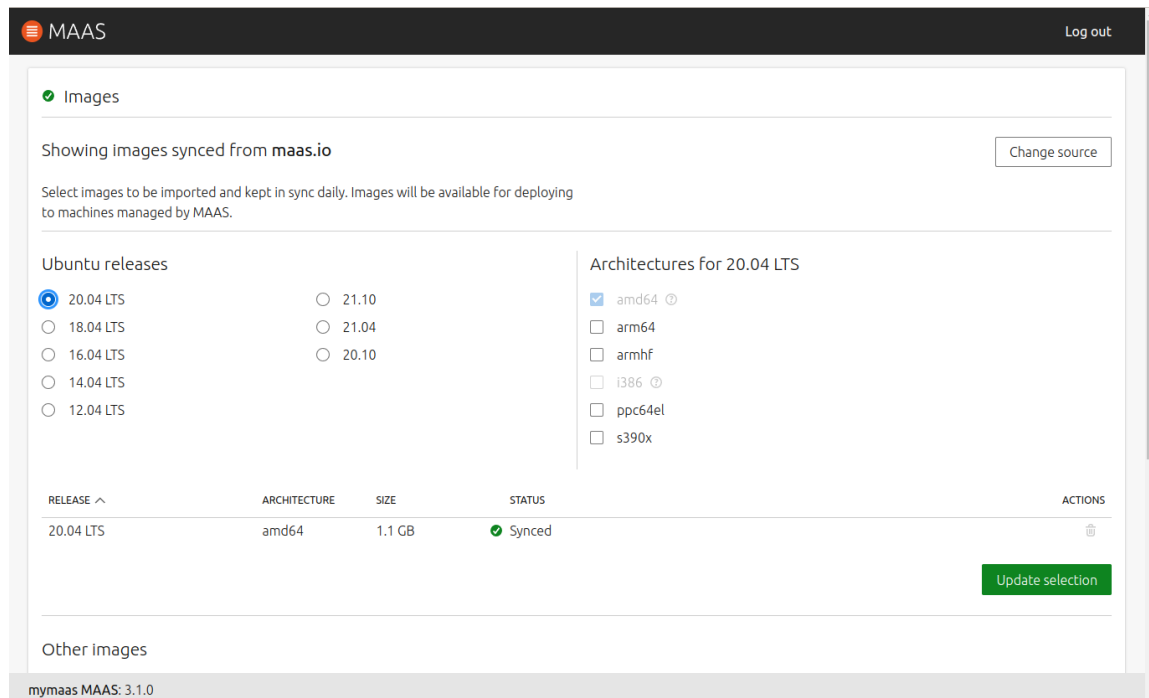
5. MAAS now displays an information page with a Finish Setup button. Click it.

6. You can now import additional SSH keys from github or Launchpad, or upload them individually. Once you've imported all the keys you need, scroll down and select Go To Dashboard. The keys that the *maniacs-setup* script imported should already be present, so you might not need to do more here. Once you're done, click Finish Setup.

7. MAAS now shows a page summarizing computers it has discovered on the network. You can review this information, but you shouldn't need to do anything on this page.

8. Click Settings in the top line, followed by Network -> DNS in the resulting sub-menu. The resulting screen should resemble the one shown below. Review your DNS settings here. The default for Enable DNSSEC Validation of Upstream Zones is Automatic, which must sometimes be adjusted. Some private DNS servers are misconfigured and will cause problems. Changing this setting to No may be required in such cases. (Alternatively, configuring the upstream DNS server to support DNSSEC should fix the problem.) If you change this option, be sure to click Save.
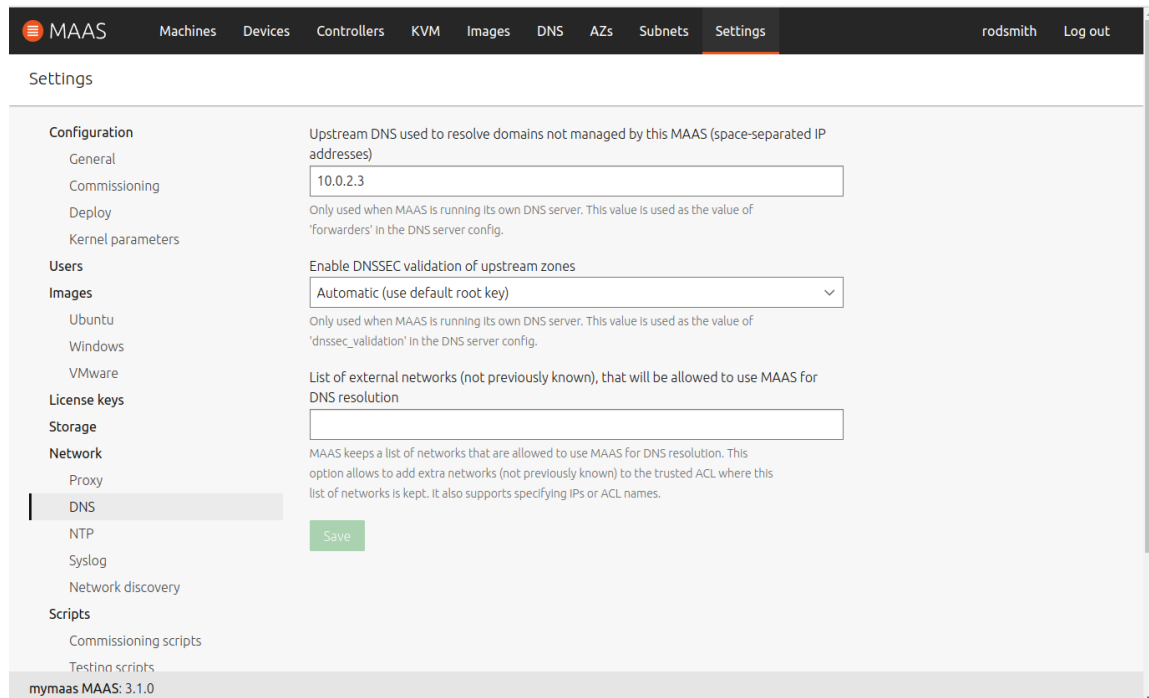
9. Select Network -> Network Discovery. In theory, this feature should passively detect devices and should cause no problems. In practice, it sometimes triggers security alerts on the external network. If you run into this problem, change this setting and click Save.

10. You can review other settings in the MAAS Settings page. This page is broken into several subsections, navigated via the list on the left of the page — Configuration, Users, Images, and so on. If you change any settings, be sure to click the associated "Save" button within that section.

11. Click the Subnets link near the top of the web page so you can review the DHCP options:

    1. Click the subnet range for the *internal* network (172.24.124.0/22 in this example):
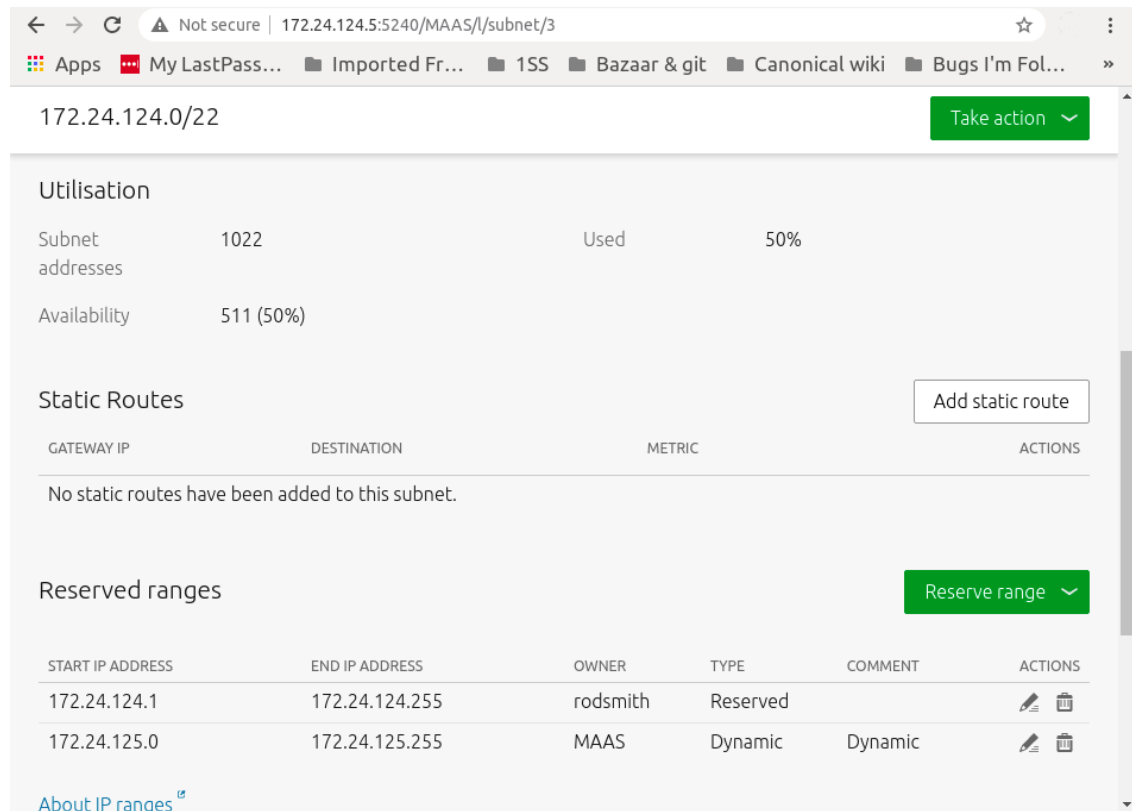
Your network, of course, may be different from this example, particularly if you have unused network devices, which will show up as additional "fabrics."

2. On the page for your internal network, scroll down about halfway to view the Utilisation and Reserved Ranges sections. At this point, about half the addresses will be classified as "used" because `maniacs-setup` set them aside as reserved or as managed by DHCP. The "available" addresses are those that do not belong to either of these categories; MAAS assigns them to nodes that are deployed using its standard settings. (See Appendix C: MAAS Network Ranges for details of how MAAS manages its network addresses.)

3. If the various ranges (reserved, dynamic, or the implicit available addresses) are not appropriate, you can edit them as follows:

- Click the edit icon near the right side of the page in the row for the range you want to delete or modify. You can then change the start and end addresses, and then click Save to save your changes.

- If you want to completely delete the range, click the trash can icon instead of the edit icon. To function properly, MAAS must have at least a small reserved range (for the MAAS server itself, at a minimum) and a dynamic range (to support enlisting, commissioning, and deploying nodes).

4. You can optionally reserve additional ranges by using the Reserve Range button, which provides two sub-options: for machines not managed by MAAS (using the Reserve Range sub-option) or for DHCP addresses (using the Reserve Dynamic Range sub-option).

Testing the MAAS Server

At this point, your MAAS server should be set up and configured correctly. (You may need to wait for images to complete importing, though; go to the Images page of the web UI to check the status of this process.) To test MAAS, follow these steps:

1. Prepare a computer by configuring it to boot via PXE. This computer need not be a computer you plan to certify; anything that can PXE-boot should work, although to fully test the MAAS server, the test system should provide IPMI or some other power-control tool that MAAS supports.

2. Connect the test computer to the MAAS server's *internal* network and power it on.

   • The test computer should PXE-boot from the MAAS server.

   • This first boot should be to the enlistment image, which provides some very basic information to the MAAS server.

   • Once the node powers itself off you should see it listed in the MAAS machines list (http://172.24.124.1:5240/MAAS/r/machines) with a Status field of "New." If it doesn't appear, try refreshing the page. Also, be sure you check the Hardware -> *Machines* tab, not any other sub-tab of Hardware.

3. Click on the node's hostname to view the node's summary page.

4. You may need to make a few changes in the Configuration area:

   • If necessary, click "Edit" in the Machine Configuration section to change the architecture of the machine. Click "Save Changes" when you're done.

   • For non-IPMI machines, you will most likely have to enter power control details by clicking Edit next to the Power Configuration heading. This may necessitate setting an IP address, MAC address, password, or other information, depending on the power control technology in use. Click "Save Changes" when you're done.

5. Click "Take Action" near the top-right corner of the page, followed by "Commission" from the resulting drop-down menu. You must then click "Start Commissioning for machine."

   • The node should power on again. This time you'll see it PXE-boot the commissioning image. Note that if your test system lacks a BMC or other means to remotely control its power, you must manually power it on.

   • The node should do a bit more work this time before powering off again.

   • Once it's done, the UI will show a Status of "Ready."

- Some servers provide an option called "minimum password change interval," or something similar, in their BMCs' web-based interfaces, that prevents BMC passwords from being changed very frequently. MAAS will attempt to change the password upon commissioning, though, and if this is done immediately after enlisting the node, it will fail. If the BMC configuration commissioning step fails, you may need to set this minimum password change interval to 0 or otherwise disable this feature, then try commissioning again. Alternatively, checking the "Skip configuring supported BMC controllers with a MAAS generated username and password" option when commissioning the node may work around this problem.

6. Once the system powers off after commissioning, click "Take Action" followed by "Deploy." You must then click "Start deployment for machine" to confirm this action.

   - The node should power on again (or you will have to control it manually if it lacks a BMC). This time it will take longer to finish working, as MAAS will install Ubuntu and the certification suite on the system.

   - Once it's done, the computer will reboot into its installed image.

   - Log into the node from the MAAS server by using SSH, as in `ssh testnode` if you've given the node the name `testnode`.

   - In the node, type `canonical-certification-precheck`. The certification suite's precheck script should run to verify that the system is ready for testing. Don't be too concerned with the results; the point of this operation is to check that the certification suite was properly installed, and at this point, additional steps are needed for the precheck script to say the node's ready for testing. These details are described in the Self-Testing Guide, which is available from your MAAS server itself, such as http://172.24.124.1.

If any of these steps fail, you may have run into a MAAS bug; your test computer may have a buggy or misconfigured PXE, IPMI, or other subsystem; or you may have misconfigured something in the MAAS setup. You may want to review the preceding sections to verify that you configured everything correctly. To help in debugging problems, the node status page includes sections entitled Commissioning, Tests, Logs, and Events with various pieces of diagnostic information related to commissioning and deployment.

At any time after enlisting a node, you can click the node's hostname near the upper-left corner of its summary page. This will enable you to change the hostname to something

descriptive, such as the computer's model number. Click "Save" when you've made your changes.

Appendix A: Adding Non-AMD64 Support

By default, the `maniacs-setup` script supports only AMD64 (64-bit, x86-64) nodes. (If you created a local mirror, it includes i386/x86 binaries because they're needed by some 64-bit packages.) Beginning with Ubuntu 20.04, direct installs in 32-bit mode to i386 computers are no longer supported, but some i386 libraries remain available to support older 32-bit binaries running on AMD64 installations. If you expect to certify computers of other architectures, such as ppc64el or ARM64, you must add support for such systems in MAAS:

1. In the MAAS web UI, click the Images tab.

2. Select the Ubuntu release you want to certify in the "Ubuntu Releases" column.

3. Select the desired CPU types in the "Architecture" column.

4. Repeat the preceding two steps for each Ubuntu release you need to certify. (You can select a different set of architectures for each Ubuntu release.)

5. Click "Update Selection." The standard MAAS images for the new CPU architectures will download. This process can take several minutes, and perhaps over an hour on a slow Internet connection.

6. For architectures other than i386 or AMD64, you must also add support for extra repositories:

    1. Click the Settings tab at the top of the MAAS web UI.

    2. Click Package Repos in the navigation pane to the left of the page.

    3. If the Ubuntu Extra Architectures repository is not enabled, click its edit icon, ensure that Enable Repository is checked, and click Save Repository.

    4. For *all* the enabled repositories (including Ubuntu Extra Architectures), click the edit icon in the Actions column, ensure that all the necessary architectures are checked, and then click Save Repository to save the changes.

That's it. Please consult the Server Certification Team if you need to certify systems using these CPUs.

Appendix B: Network Testing Options

A key part of certification is testing your SUT's network cards. This document is written with the assumption of a fairly basic configuration; however, some labs may have more advanced needs. Important variables include:

- **Multiple simultaneous network tests** — A single server takes about 60 minutes per network port to run its network tests — long enough that testing multiple SUTs simultaneously is likely to result in contention for access to the `iperf3` server. This is especially true if SUTs have multiple network ports — a server with four ports will tie up an `iperf3` server for four hours. An `iperf3` server will refuse multiple connections, which should at least enable one SUT's network tests to pass; but if the `iperf3` server has a sufficiently fast NIC, it will then be under-utilized.

- **Advanced network interfaces** — A low-end computer configured as described here will likely have a 1 Gbps link to the internal LAN. If you're testing systems with faster interfaces, you will need a separate computer to function as an `iperf3` server.

If your `iperf3` target system has a fast NIC and you want to test multiple slower SUTs, you can configure the fast NIC with multiple IP addresses. A NetPlan configuration (as used in Ubuntu 17.10 and later) to support multiple IP addresses can be enabled in `/etc/netplan/50-cloud-init.yaml` (or another file; the name varies depending on how the system was installed). For example:

```
eno2:
    addresses:
    - 172.24.124.2/22
    - 172.24.124.3/22
    - 172.24.124.4/22
```

Note that you do not explicitly set separate names for each interface.

You must activate the changes after making them. In theory, you can do this without rebooting by typing `sudo netplan apply`; however, you may find it's necessary to reboot to reliably apply an advanced configuration like this one. You can verify the network settings with `ip addr show eno2` (changing the interface name as necessary):

```
$ ip addr show eno2
3: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state
```

```
 UNKNOWN group default qlen 1000
link/ether 08:00:27:90:0e:07 brd ff:ff:ff:ff:ff:ff
inet 172.24.124.2/22 brd 172.24.127.255 scope global eno2
   valid_lft forever preferred_lft forever
inet 172.24.124.3/22 brd 172.24.127.255 scope global secondary eno2
   valid_lft forever preferred_lft forever
inet 172.24.124.4/22 brd 172.24.127.255 scope global secondary eno2
   valid_lft forever preferred_lft forever
inet6 fe80::a00:27ff:fe90:e07/64 scope link
   valid_lft forever preferred_lft forever
```

This example shows eno2 up with all three of its IP addresses. Note that the older ifconfig tool will show only the first IP address for any device configured via NetPlan.

You would then launch iperf3 separately on each IP address:

```
iperf3 -sD -B 172.24.124.2
iperf3 -sD -B 172.24.124.3
iperf3 -sD -B 172.24.124.4
```

On the MAAS server, you can enter all of the iperf3 target addresses in /etc/maas-cert-server/iperf.conf:

```
172.24.124.2,172.24.124.3,172.24.124.4
```

The result should be that each of your SUTs will detect an open port on the iperf3 server and use it without conflict, up to the number of ports you've configured. Past a certain point, though, you may over-stress your CPU or NIC, which will result in failed network tests. You may need to discover the limit experimentally.

Furthermore, if you want to test a SUT with a NIC that meets the speed of the iperf3 server's NIC, you'll have to ensure that the high-speed SUT is tested alone — additional simultaneous tests will degrade the performance of all the tests, causing them all to fail.

If the iperf3 server has multiple interfaces of differing speeds, you may find that performance will match the *lowest-speed* interface. This is because the Linux kernel arbitrarily decides which NIC to use for handling network traffic when multiple NICs are

linked to one network segment, so the kernel may use a low-speed NIC in preference to a high-speed NIC. Two solutions to this problem exist:

- You can disable the lower-speed NIC(s) (permanently or temporarily) and rely exclusively on the high-speed NIC(s), at least when performing high-speed tests.

- You can configure the high-speed and low-speed NICs to use different address ranges — for instance, 172.24.124.0/22 for the low-speed NICs and 172.24.128.0/22 for the high-speed NICs. This approach will require additional MAAS configuration not described here. To minimize DHCP hassles, it's best to keep the networks on separate physical switches or VLANs, too.

If your network has a single `iperf3` server with multiple physical interfaces, you can launch `iperf3` separately on each NIC, as just described; however, you may run into a variant of the problem with NICs of differing speed — the Linux kernel may try to communicate over just one NIC, causing a bottleneck and degraded performance for all tests. Using multiple network segments or bonding NICs together may work around this problem, at the cost of increased configuration complexity.

If your lab uses separate LANs for different network speeds, you can list IP addresses on separate LANs in `/etc/maas-cert-server/iperf.conf` on the MAAS server or in `/etc/xdg/canonical-certification.conf` on SUTs. The SUT will try each IP address in turn until a test passes or until all the addresses are exhausted.

If you want to test multiple SUTs but your network lacks a high-speed NIC or a system with multiple NICs, you can do so by splitting your SUTs into two equal-sized groups. On Group A, launch `iperf3` as a server, then run the certification suite on Group B, configuring these SUTs to point to Group A's `iperf3` servers. When that run is done, reverse their roles — run `iperf3` as a server on Group B and run the certification suite on Group A. You'll need to adjust the `/etc/xdg/canonical-certification.conf` file on each SUT to point it to its own matched server.

Testing high-speed network devices (above 10 Gbps) requires changing some network configuration options. Appendix D of the Ubuntu Server Certified Hardware Self-Testing Guide covers how to configure both the SUT and the `iperf3` Target system for such testing. Configuration of one feature in particular, though, can be facilitated via MAAS: jumbo frames. When testing servers with high-speed network interfaces (those over about 10Gbps), it's often necessary to set jumbo frames (an MTU significantly higher than 1500, and typically 9000) on the iperf3 server, the SUT, and any intervening switches. By default, MAAS configures SUTs with an MTU of 1500; however, you can change this detail by editing the MAAS settings:

1. On any MAAS web UI screen, click Subnets.

2. On the resulting page, locate the fabric corresponding to your high-speed network and click the link under the VLAN column. (This is usually entitled "untagged," unless you've configured VLAN tagging.)

3. Under VLAN Summary, you'll probably see the MTU as set to 1500. If it's already set to 9000, you don't need to make any changes. If it's 1500, though, click the Edit button near the top-right of this section.

4. The resulting input fields enble you to change configuration details for this network. Change the MTU field to 9000.

5. Click Save Summary to save the changes.

6. Perform a test deployment and verify that the node's MTU is set to 9000 for the interface(s) connected to the high-speed network.

You can make this change even to lower-speed networks or to networks with mixed speeds; however, the change applies to *all* the computers that MAAS controls on the associated fabric. Because jumbo frames create problems in some cases (such as PXE-booting some older UEFI-based computers), you should be cautious about applying this change too broadly. That said, if it works for your servers, there's little reason to *not* set jumbo frames universally. Note that this change will not automatically adjust your iperf3 servers' MTUs, so you may need to set them manually, as described in the Self-Test Guide. You may also need to adjust your switches, since they must support jumbo frames, too, in order to get their speed benefit.

You may find the `iftop` utility helpful on the `iperf3` server system. This tool enables you to monitor network connections, which can help you to spot performance problems early.

Appendix C: MAAS Network Ranges

As noted earlier, in Installing and Configuring Ubuntu, a /22 or wider network on the internal port is desirable, because this provides more addresses that are assigned more flexibly than with smaller networks. Specifically, MAAS splits the internal network into three parts:

- A reserved space, from which you can assign addresses manually. The MAAS server itself should be in this space. You might also use this space for other permanent infrastructure on the network, such as switches or other necessary servers. If you assign static IP addresses to your BMCs, their addresses would either come out of this space or be on another network block entirely.

- A dynamic (DHCP) space, which MAAS manages so that it can temporarily address servers when enlisting and commissioning them. Depending on your needs, your BMCs and even deployed nodes may be assigned via DHCP, too.

- An automatic space, which is a range of addresses that MAAS assigns to nodes. MAAS configures these nodes to use static addresses, not DHCP; but although the static addresses will survive reboots, they are likely to change between deployments.

In the MAAS subnet configuration page, the reserved and dynamic spaces are explicitly defined. Any address that does not fall into either of those spaces is part of the automatic space.

The following table shows how the `maniacs-setup` script described in this document splits up a /22, a /23, and a /24 network, starting with 172.24.124.1, between these three purposes. You can adjust the ranges after they've been set up by using the MAAS web UI, as described earlier, in Checking the MAAS Configuration, should the need arise. If you use a network block starting at something other than 172.24.124.1, the exact IP addresses shown in the table will be adjusted appropriately.

Purpose	/22 network	/23 network	/24 network
Reserved	172.24.124.1 - 172.24.124.255	172.24.124.1 - 172.24.124.50	172.24.124.1 - 172.24.124.9
Dynamic	172.24.125.0 - 172.24.125.255	172.24.124.51 - 172.24.124.255	172.24.124.10 - 172.24.124.127
Assigned Automatically	172.24.126.0 - 172.24.127.254	172.24.125.0 - 172.24.125.254	172.24.124.128 - 172.24.124.254

Appendix D: Installing MAAS in a LXD Container

It is possible to install MAAS in a virtual machine or container. Doing so will help to isolate MAAS from the underlying OS and enable relatively easy backup and restoration of the complete MAAS environment. To facilitate this setup, the certification PPA includes a package, called `maas-lxc-host`, which includes scripts and tools to run on the host system in order to install MAAS in a LXC/LXD container. Note, however, that *this procedure is still experimental!* Although it can be made to work, it is delicate, and can easily fail because of system-specific configuration issues or because of minor deviations from the specified procedure.

If you want to run MAAS in this way, follow these steps:

1. Ensure that the host has sufficient disk space. The container consumes 128 GiB of disk space, in `/var/snap/lxd/common/lxd/disks/`.

2. Install Ubuntu Server on the server you want to host the LXD container and configure the server's network as described earlier, in Installing and Configuring Ubuntu.

3. Type `sudo apt-add-repository ppa:hardware-certification/public` to add the Hardware Certification PPA to the host server.

4. Install the `maas-lxc-host` package by typing `sudo apt install maas-lxc-host`.

5. If you're using a remote SSH session, type `screen`. The setup process may interrupt network connectivity, so you'll have to reconnect mid-process. Better, use a physical console or remote KVM, which will not be affected by this interruption.

6. Type `lxc-setup`. This runs the LXC/LXD setup script, which proceeds to run through the setup steps, asking you some questions along the way…

   1. If you're running remotely, the script checks to see if `screen` is in use. If so, you'll be asked to confirm that you want to continue.

   2. At least once, and perhaps multiple times, you'll be asked to enter your password. Do so whenever prompted.

   3. The script tries to identify the internal and external network devices on the host, and asks you to verify each one. It then creates network bridges for the future container. Once this is done, the script gives you the option to manually edit the NetPlan configuration file, in case you want to make your own tweaks. Note that the script tries to configure the external network interface

(`br1`) using DHCP. This is likely to result in the external network interface's IP address changing compared to its original configuration unless you manually edit it to use a static IP address; but this may not be appropriate. You should make changes suitable for your own network.

4. After configuring the network, your remote network access is likely to go down, if you're running remotely. You should be able to reconnect (doing so via the internal network interface may be easier than trying to find the new external IP address) and run `screen -r` to resume.

5. After configuring LXD, the script sets up the LXD container's network options. As with the host's network configuration, the script gives you the option of reviewing and editing the settings. By default, the internal network (`eth0`) is given an address one higher than the host (for instance, 172.24.124.2, to the host's 172.24.124.1), and the external address is configured via DHCP.

6. At this point, the script asks if you want to set up MAAS in the LXD container. If you respond by typing `Y` (which is the default), the script installs `maas-cert-server` in the LXD container and then runs `maniacs-setup`, as described in Running the Setup Script. For the most part, you can configure the MAAS server in the LXD container just as you would a MAAS server running directly on a server; however....

7. When the setup script asks if you want to configure the server as a NAT router, you may want to answer `N`. The NAT control scripts are installed on both the host and the container, but using the host as a NAT router provides a more direct route to the outside world than would be the case if you used the LXD container for this purpose. this topic is covered in more detail shortly.

7. If the `lxc-setup` script fails at some point, you can try fixing whatever problem is reported and re-running the script; however, this use case is not yet well-tested and so may fail. You may need to copy `/usr/sbin/lxc-setup` to your home directory and edit it to work around the problems.

The result of this configuration is that the computer will have at least four IP addresses: internal and external for the host computer itself and for the LXD container. The latter will run MAAS and an SSH server, but the latter may not be usable until you import your SSH public keys into the LXD container's `ubuntu` account. You can do this from the host by typing:

```
lxc exec lxc-maas bash
su ubuntu
ssh-import-id lp:username
exit
exit
```

Change `username` to your Launchpad username. Alternatively, you can add SSH public keys in any way you like, such as by editing `~/.ssh/authorized_keys`.

If you prefer, you can access the LXD container from the host by typing `lxc exec lxc-maas bash` every time; however, this is likely to be more awkward than enabling direct SSH access to the container.

In either case, the LXD container shares the `/home/username` and `/srv` directories with the host, where `username` is your username on the host. The former enables you to easily share arbitrary files between the host and its container; and the latter is intended to simplify configuration of Apache to deliver virtualization files needed by the virtualization tests. Because `maniacs-setup` configures Apache on the LXD container and optionally downloads virtualization files, it's easy to set up the container as the server for these files; however, storing these large files outside of the container may be desirable. You can also install Apache on the host and deliver these files from that location, if you prefer.

The `maniacs-setup` script configures the MAAS server computer (that is, the LXD container, when MAAS is installed this way) as the router for the internal network. If you want to use the host instead, you must take some extra steps:

1. On the host computer, type `sudo systemctl enable certification-nat` to configure it to enable NAT on the next reboot

2. Type `sudo service certification-nat start` on the host to start NAT immediately.

3. In the MAAS web UI, select Subnets from the options at the top of the page, and then select the internal subnet (under the "Subnet" column) from the list.

4. Click the Edit button to the right of the Subnet Summary section.

5. Change Gateway IP to match the host computer's IP address, rather than the LXD container's IP address. (You can make other changes here, too, if necessary for your network.)

6. Click Save Summary to save your changes.

The `iperf3` server is installed on both the host and the LXD container at the end of this process, but it's not configured to launch automatically from either location. In theory, network tests can use either location as a target, once you launch `iperf3` in the correct environment; however, running `iperf3` on the host is less likely to cause performance problems and is therefore recommended. If you run `iperf3` on the LXD container and encounter network test failures, you should try running `iperf3` on the host and using it as a target instead. Note that network performance will be limited by the capabilities of the host; you can't run full-speed tests against both the host and the LXD container and expect to get twice the host's native network speed!

A fresh installation of MAAS in a LXC/LXD container will consume about 6 GiB of disk space in the `/var/snap/lxd/common` directory (or `/var/lib/lxd/storage-pools/default/containers/`, if using an Ubuntu 18.04 host). This space is likely to grow over time, especially if you add support for multiple Ubuntu versions and CPU architectures to your MAAS configuration. (Each new version requires `cloud-init` files that consume some space.)

You can use numerous commands to manage your MAAS container. These include, but are not limited to:

- `lxc list` — Shows a list of containers and some summary information about them, including their IP addresses and whether or not they're running.

- `lxc info` — Displays summary information about a specified container (more than is shown by `lxc list`).

- `lxc exec` — Runs a command in a container. In particular, `lxc exec lxc-maas bash` runs bash in the `lxc-maas` container (the name of the container created by `lxc-setup`).

- `lxc stop` — Stops a specified container.

- `lxc start` — Starts a specified container. Note that the container created by `lxc-setup` should start up automatically when the host boots.

- `lxc restart` — Restarts a specified container.

- `lxc snapshot` — Creates a snapshot of a specified container.

- `lxc restore` — Restores a snapshot of a specified container.

The `lxc-setup` script creates a container that's 128 GiB in size. This is normally adequate. (If you create a local APT mirror, that mirror can be much bigger than this, but it will normally be hosted in `/srv`, which is a filesystem that's shared with the host, and so does not count against the container's size.) Versions of `lxc-setup` prior to `maas-cert-server`

0.6.2, however, created a container that's only 30 GiB in size. If the container fills up, symptoms can include a sluggish container, an unresponsive MAAS server, and a high CPU load on the host. You can type `df /` inside the container to check its disk use. If you find the container is low on disk space, you may want to begin by reviewing your installed images in MAAS. Delete unused images, such as for old releases or architectures you don't test. If you're still low on disk space in the container, you can increase its size as follows:

1. On the host, verify that `/var/snap/lxd/common/lxd/disks/default.img` exists. This file should hold the container's filesystem; but its location could differ if you installed in some unusual way or if you're using something other than 20.04 as the host OS.

2. On the host, check to see how much disk space is available in the filesystem that holds the container, as just identified. (This is usually in your root filesystem, /, so `df -h /` will give you the information you need.)

3. On the host, type the following commands:

```
sudo truncate -s +100G /var/snap/lxd/common/lxd/disks/default.img
sudo zpool set autoexpand=on default
sudo zpool online -e default /var/snap/lxd/common/lxd/disks/default.img
sudo zpool set autoexpand=off default
```

   If necessary, change the path to the container's filesystem file; and if desired or necessary, change `+100G` to a suitable value for a change to the filesystem size. *Be sure that* `100G` *is preceded by a plus sign (+)!*

4. Log into the container.

5. Verify that the available disk space has increased, such as by typing `df -h /`.

This documentation can provide only a brief summary of LXC/LXD commands and tools. For more information, see the official Linux containers documentation at https://linuxcontainers.org. You can also type `lxc` with no options to see a summary of sub-commands, or type `lxc` with a subcommand to see a summary of how to use it, if the subcommand requires additional options.

Appendix E: Glossary

The following definitions apply to terms used in this document.

1 Gbps

    1 Gigabit - Network speed for Gigabit Ethernet (1000 Mbps).

10 Gbps

    10 Gigabit - Network speed for 10 Gigabit Ethernet (10,000 Mbps).

BMC

    Baseboard Management Controller — A device in many server models that allows remote in- and out-of-band management of hardware.

DHCP

    Dynamic Host Control Protocol — method for providing IP addresses to the SUTs.

IPMI

    Intelligent Platform Management Interface — A technology for remotely connecting to a system to perform management functions.

LAN

    Local Area Network — the network to which your SUTs are connected. The LAN does not need to be Internet accessible (though that is preferable if possible).

MAAS

    Metal as a Service — a Canonical product for provisioning systems quickly and easily.

NIC

    Network Interface Card — the network device(s).

PXE

    Pre-boot Execution Environment — A technology that allows you to boot a system using remote images for easy deployment or network-based installation.

SUT

    System Under Test — The machine you are testing for certification.