

This tutorial shows the installation of an Ubuntu 20.04 LTS base server in detail with many screenshots. The purpose of the guide is to show the installation of Ubuntu 20.04 LTS that can be used as the basis for our other Ubuntu tutorials here at howtoforge like our perfect server guides. This tutorial uses the LTS branch which gets updates for 5 years from Ubuntu and is recommended for use on servers.

1. Requirements

To install an Ubuntu Server, you will need the following prerequisites:

- The Ubuntu 20.04 server ISO image, available here: <https://releases.ubuntu.com/20.04/ubuntu-20.04-live-server-amd64.iso> (For 64Bit Intel and AMD CPU's). Take a look here for other Ubuntu downloads: <https://releases.ubuntu.com/20.04/>
- A fast internet connection is recommended as the package updates get downloaded from Ubuntu servers during installation.

2. Preliminary Note

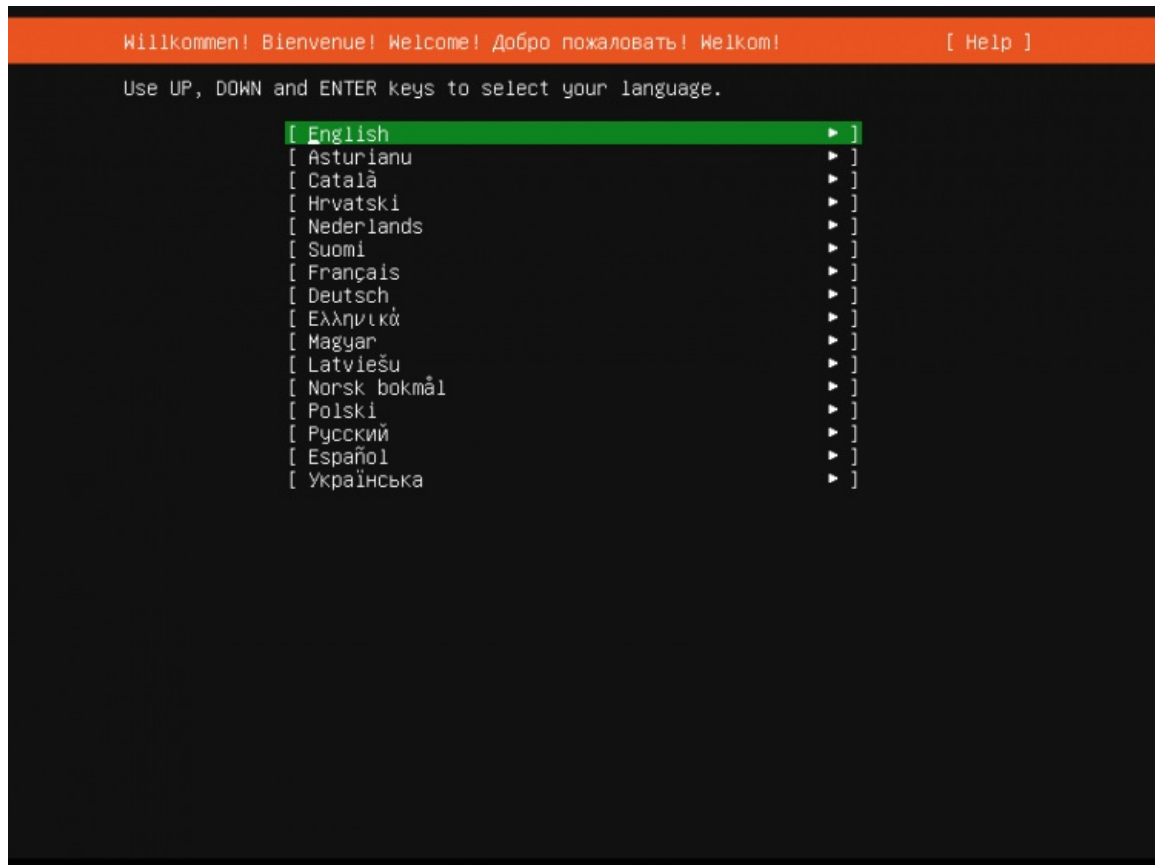
In this tutorial, I use the hostname *server1.example.com* with the IP address *192.168.0.100* and the gateway *192.168.0.1* These settings might differ for you, so you have to replace them where appropriate.

3. Installing the Ubuntu 20.04 Base System

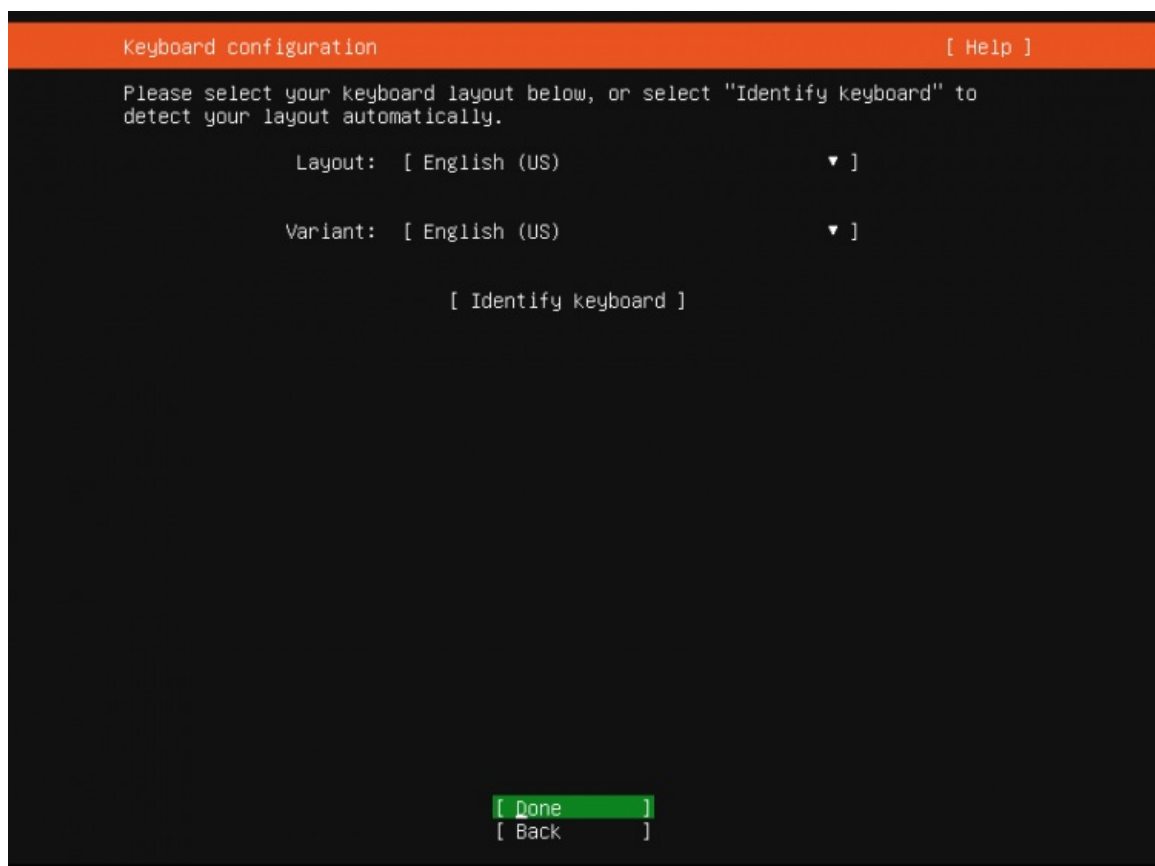
Insert the Ubuntu install CD / DVD / USB stick into your system and boot from it. When you install the OS in a virtual machine like I will do it here, then you should be able to select the downloaded ISO file as a source for the CD/DVD drive in VMWare and Virtualbox without burning it on CD first. Start the server or virtual machine, it will boot the system and start the installer.

```
[ 3.002504] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
[ 3.587514] sd 2:0:0:0: [sda] Assuming drive cache: write through
ln: /tmp/mountroot-fail-hooks.d//scripts/init-premount/lvm2: No such file or directory
Checking integrity, this may take some time
....mount: mounting /cow on /root/cow failed: No such file or directory
Connecting to plymouth: Connection refused
....._
```

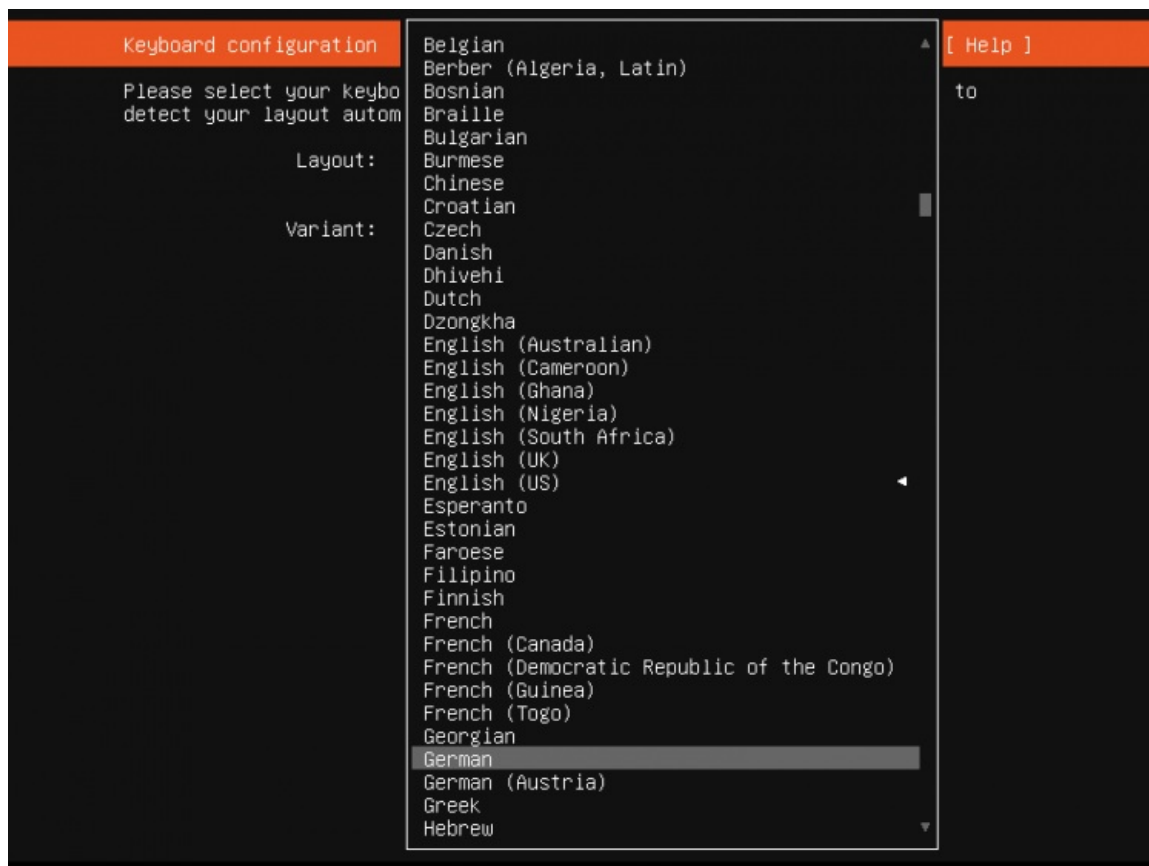
The first screen of the Ubuntu installer will show the language selector. Please select your language for the installation process:



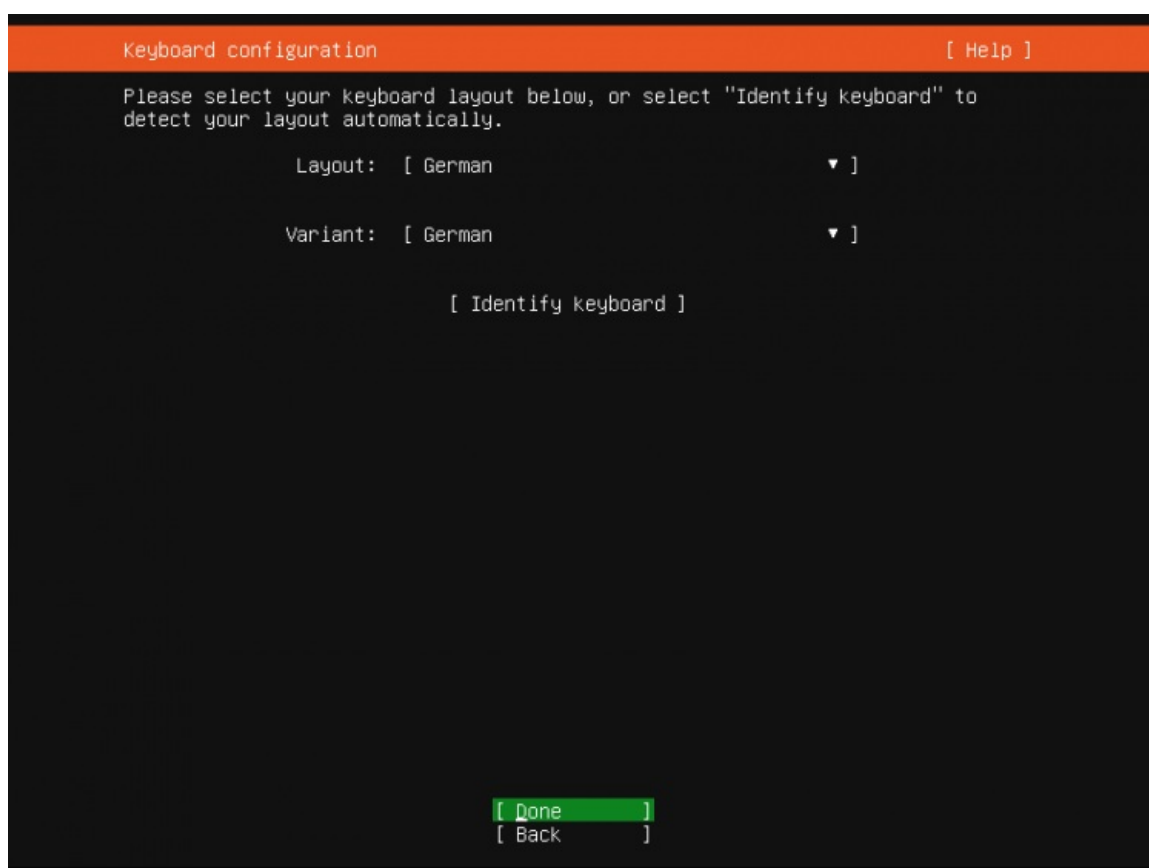
On the next screen, you can choose the keyboard layout. The English keyboard will be fine for many users. In this case, choose "Done" at the end of the screen and press the "Return" key, to go to the next step. In my case, I'm using a German keyboard layout, this means I'll have to navigate to the "Layout" option by pressing the "Tab" key on my keyboard until the **Layout** option is highlighted. Then press the "Return" key to open the Layout selector.



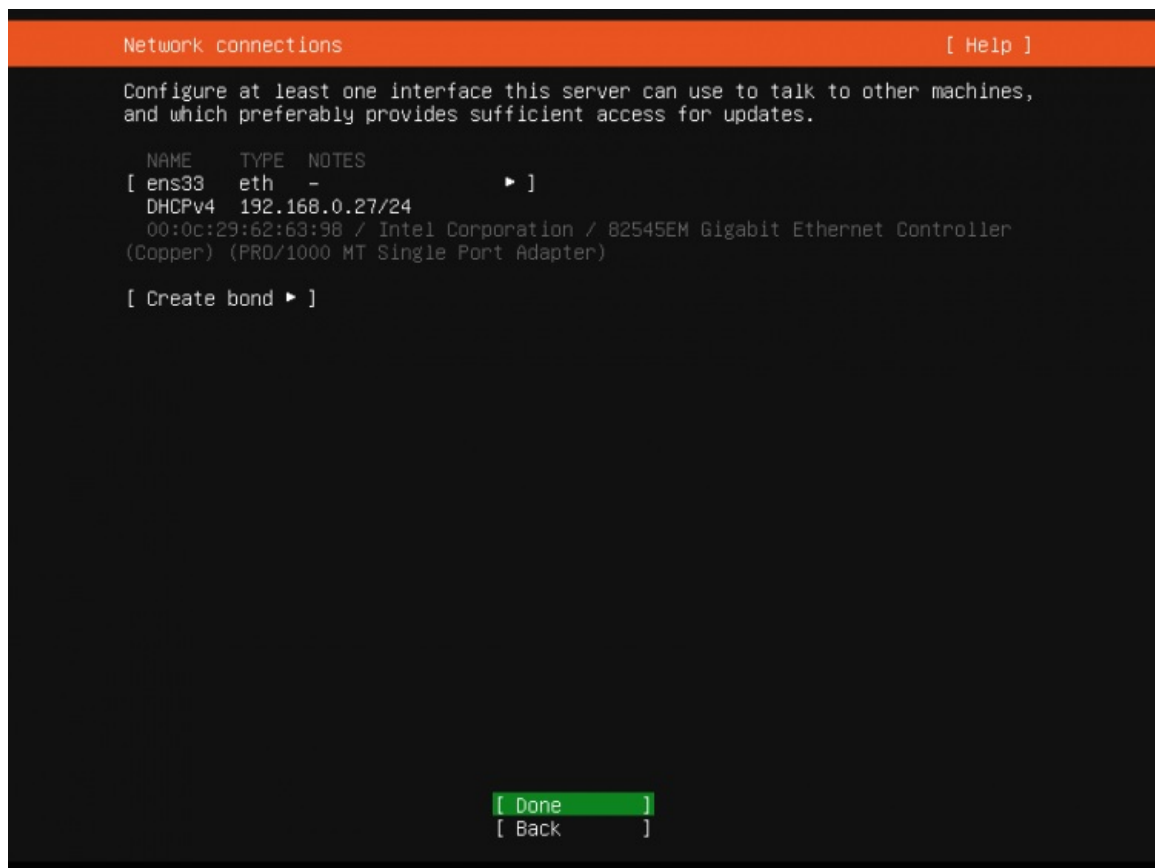
Choose the keyboard layout that matches the keyboard that is attached to the server.



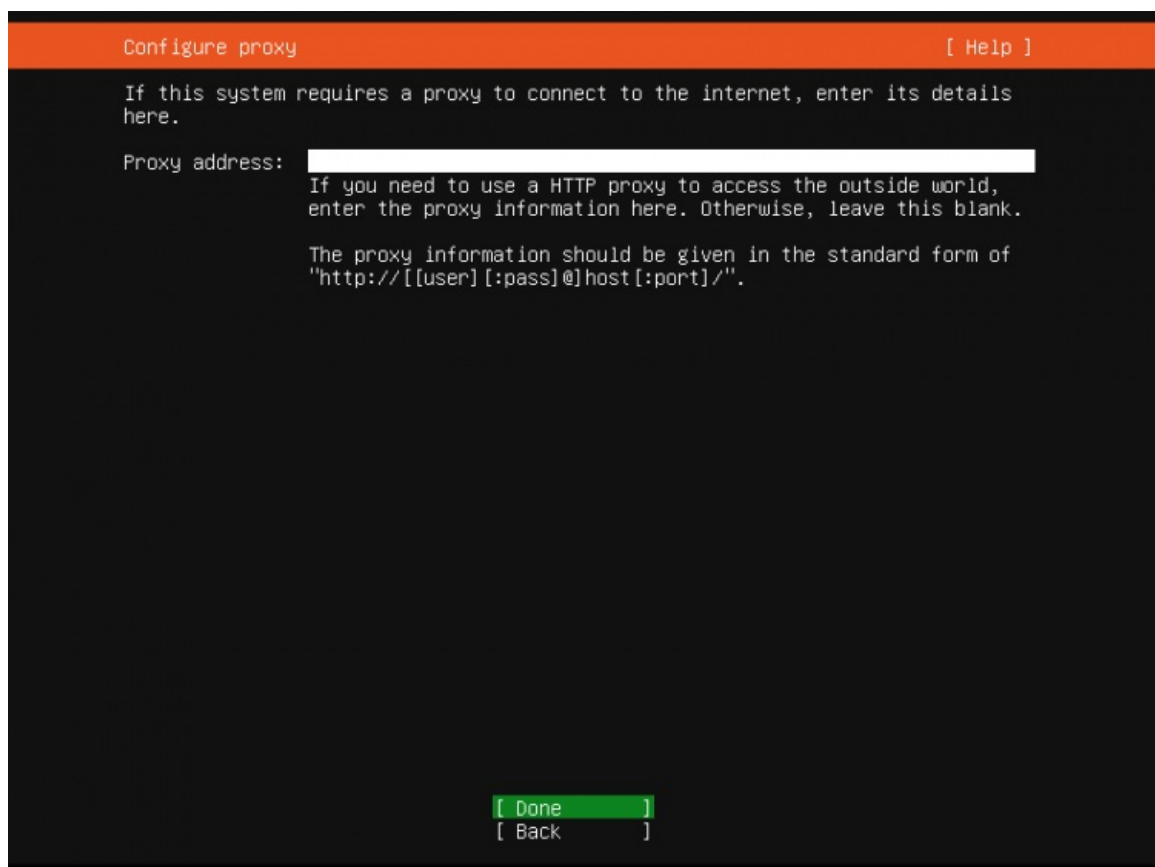
The right keyboard layout for my installation is selected now. Choose "Done" at the end of the screen and press "Return", to go to the next step.



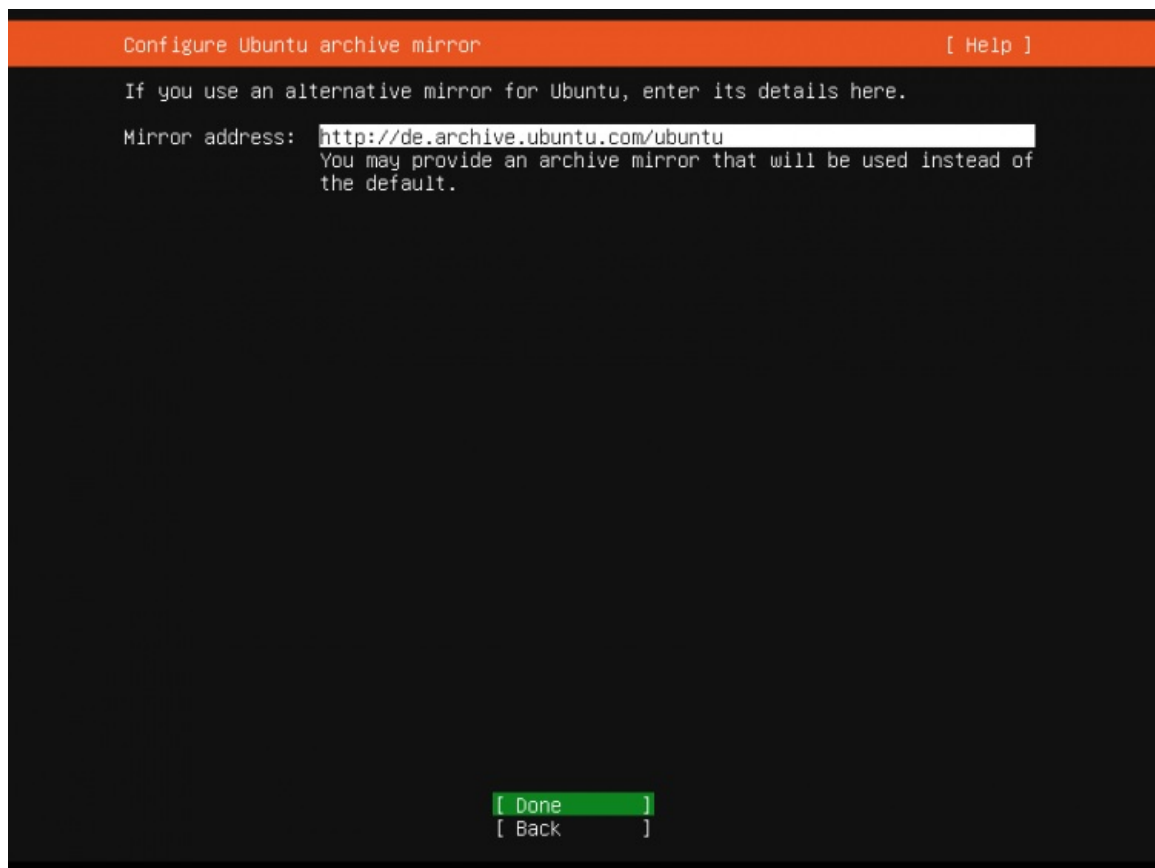
The Ubuntu installer shows now which network card it has detected on the server. The network device name which got assigned automatically is ens33. The IPv4 address has been assigned automatically via DHCP. I will change it later to a fixed IP address when the base system has been installed. If your network has no DHCP server, then you can enter a fixed IP address now by choosing the network card (press Tab until it is highlighted and then press Return).



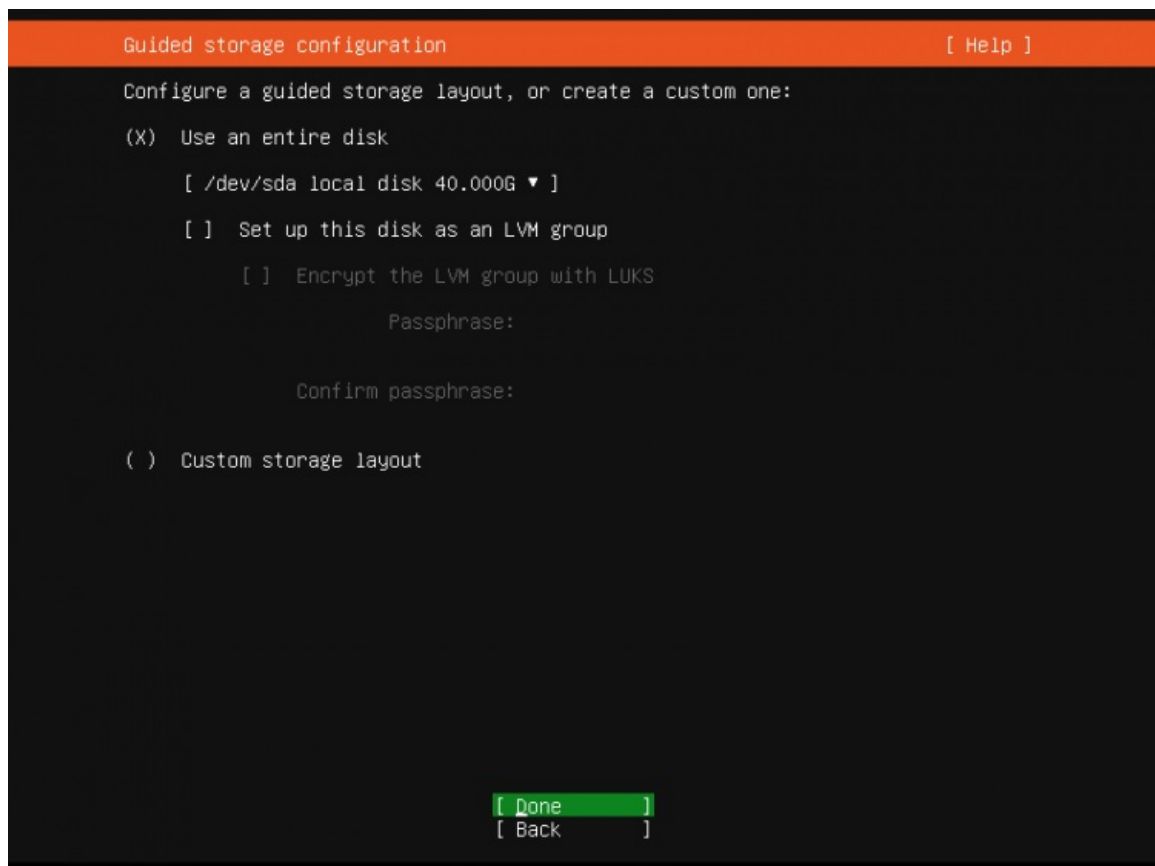
Now you can set a proxy server address in case a proxy is required to access the internet. In my case, there is no proxy required, so I just choose "Done" to go to the next installation step.



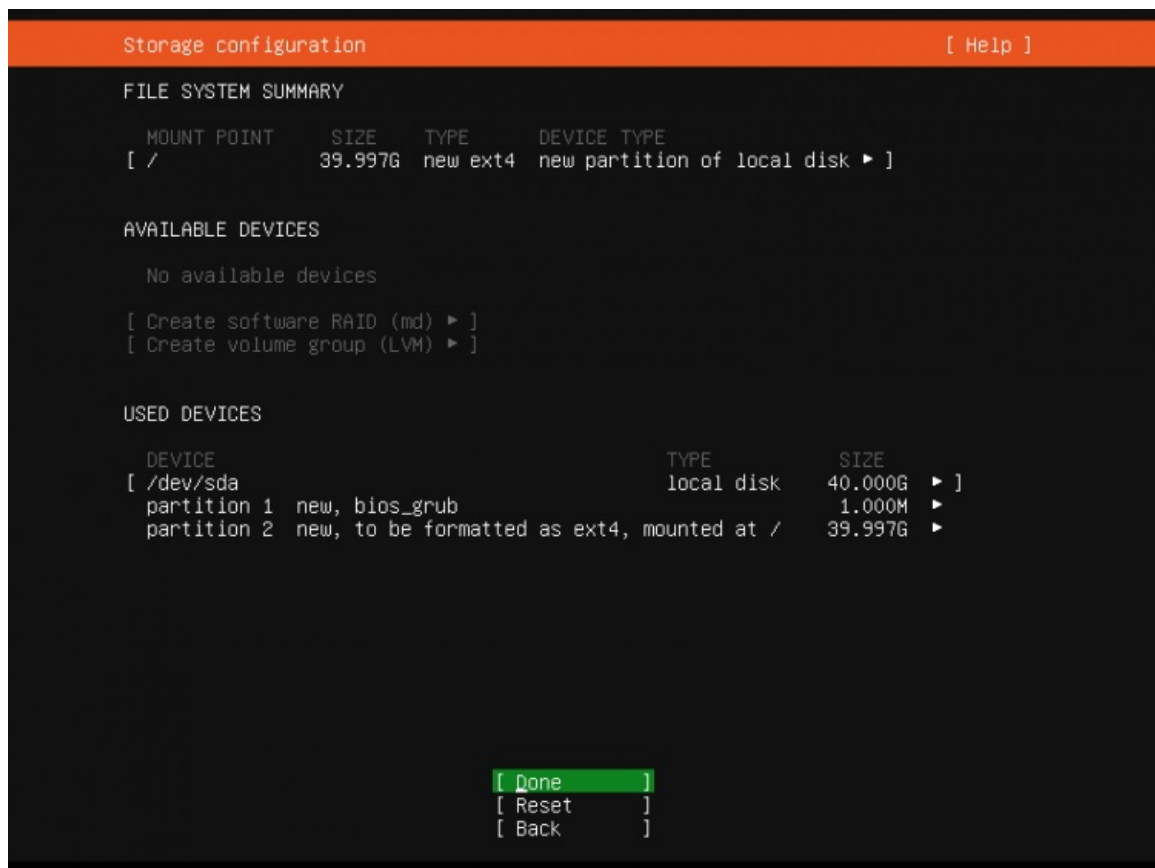
Here you can choose from which Ubuntu mirror server updates and installation files shall be downloaded. I'll keep the default and go to the next install screen.



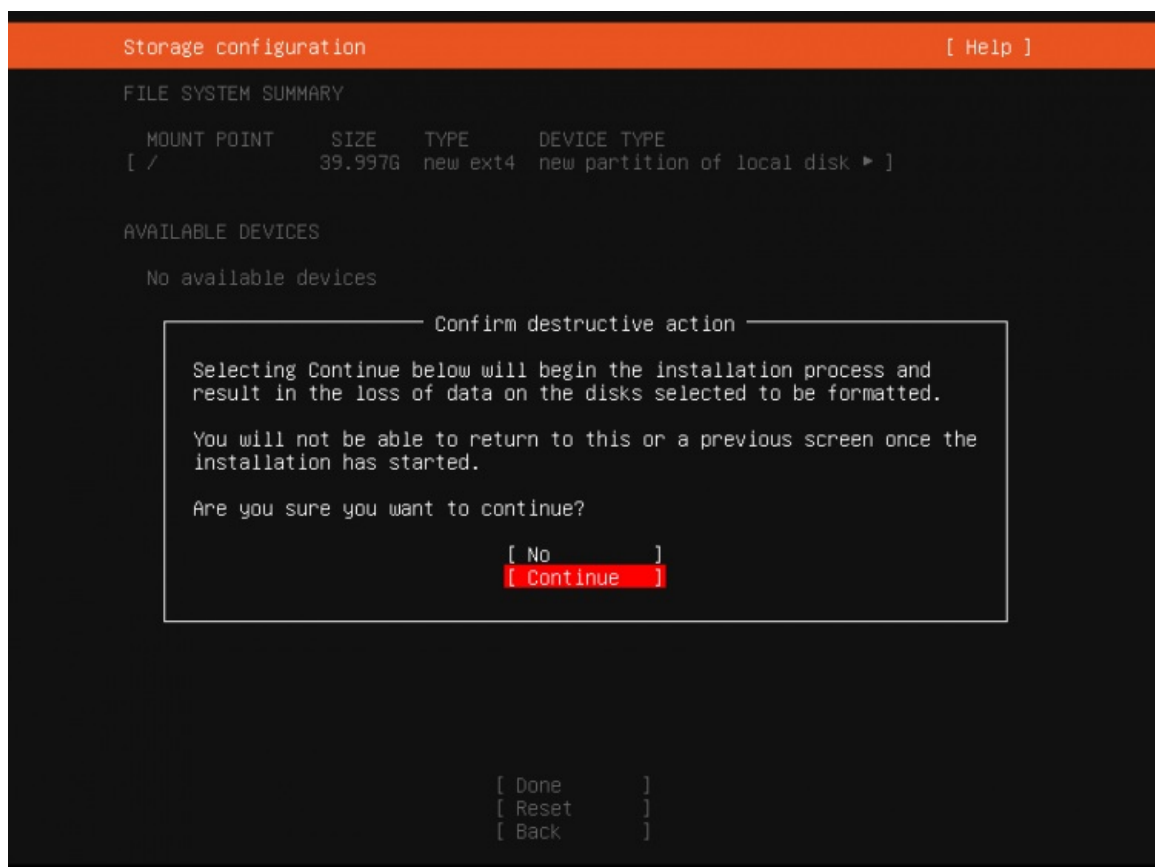
The Ubuntu server installer shows now the harddisks that it has detected in the server. The installation disk is a 40GB HD on /dev/sda here. I'll choose to use the entire disk for my Ubuntu installation. If you need a custom layout consisting of multiple partitions, choose "Custom Layout" instead and create partitions as needed.



The installer shows the default storage configuration below. It consists of a 1MB bios_grub partition plus one large / partition that will contain the operating system installation. Choose "Done" to proceed to the next screen.



Before the installation starts, the Ubuntu installer requests to confirm the partitioning. Press the "Tab" key until the "Continue" option is highlighted in red, then press "Return" to proceed.



Now it's time to set the server name (hostname) and the username and password of the administrator. I'll choose the username 'administrator' here just as an example, please use a different and more secure name in your real setup. The Ubuntu shell user that we create in this step has sudo permissions, this means that he is able to administrate the system and to become root user via sudo.

Profile setup

[Help]

Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.

Your name:

Your server's name:
The name it uses when it talks to other computers.

Pick a username:

Choose a password:

Confirm your password:

[Done]

Most Linux servers get administered over the network using SSH. In this step, the Ubuntu installer can install the SSH server directly. Select the checkbox "Install OpenSSH Server" and proceed to the next step.

SSH Setup

[Help]

You can choose to install the OpenSSH server package to enable secure remote access to your server.

☒ Install OpenSSH server

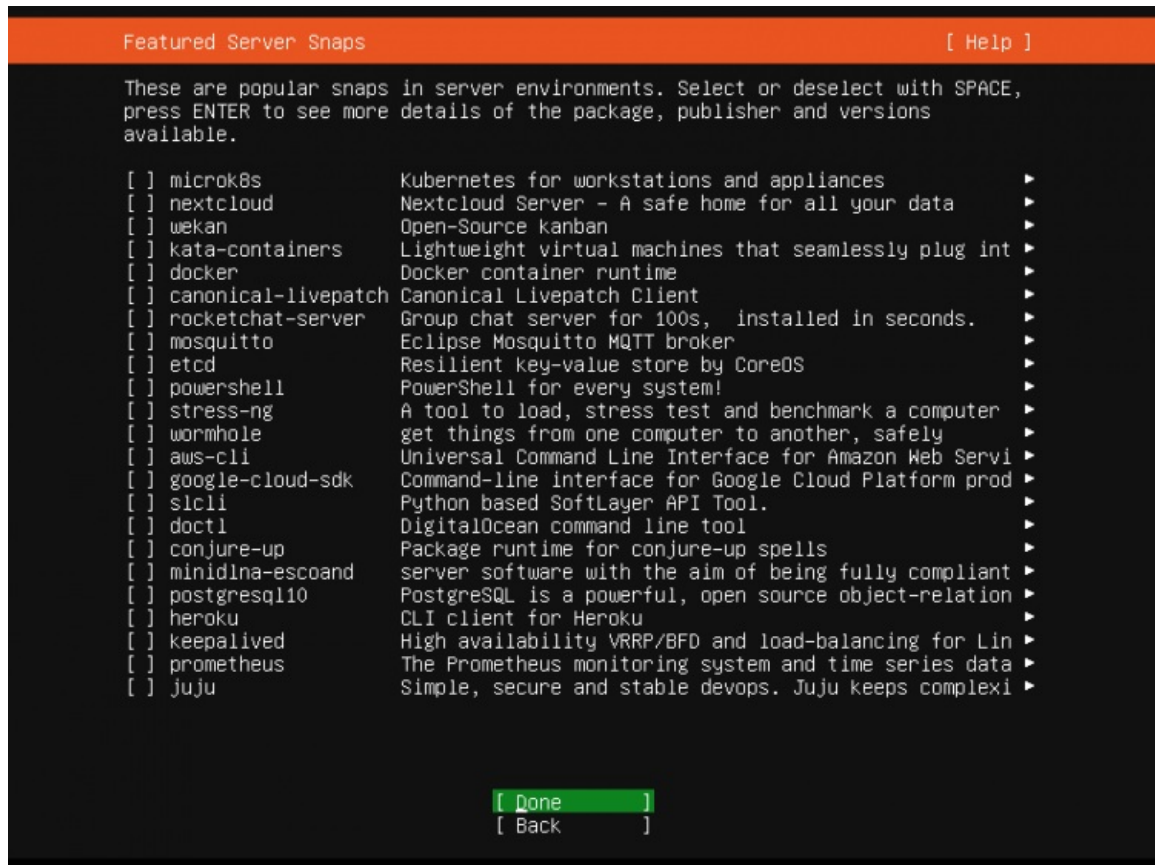
Import SSH identity:
You can import your SSH keys from Github or Launchpad.

Import Username:

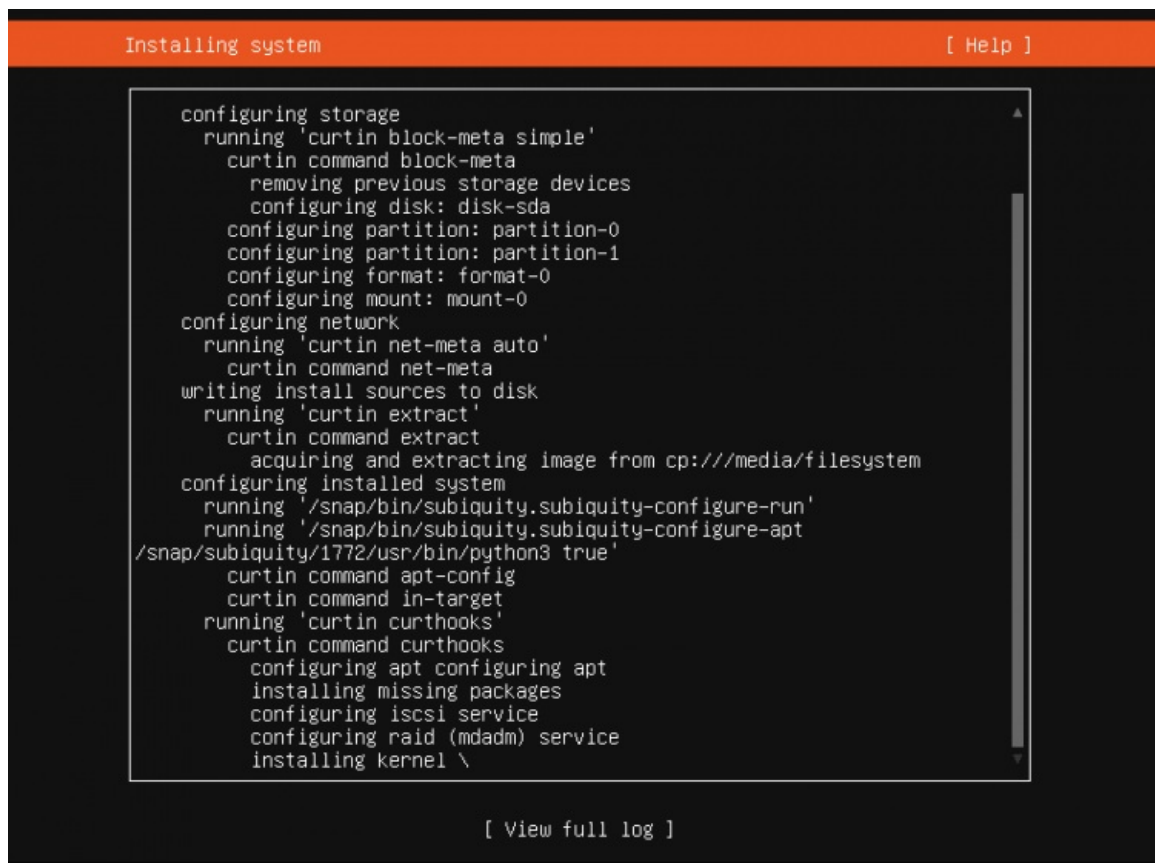
☒ Allow password authentication over SSH

[Done]
[Back]

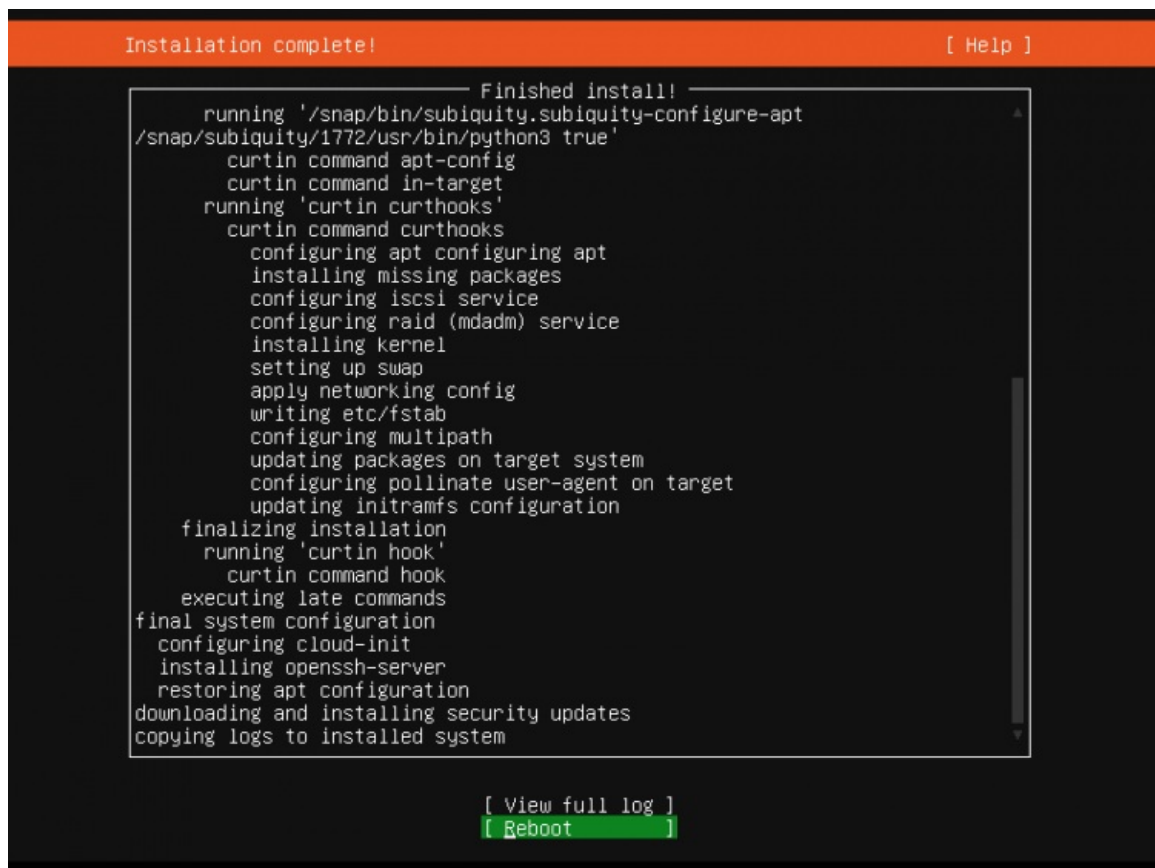
In this step, you can preinstall commonly used services via Snap installer. I do not select any services here as the purpose of this guide is to install a minimal base system. You can install services via apt or snap at any time later.



The Ubuntu installer now proceeds with the installation based on the settings we have chosen.



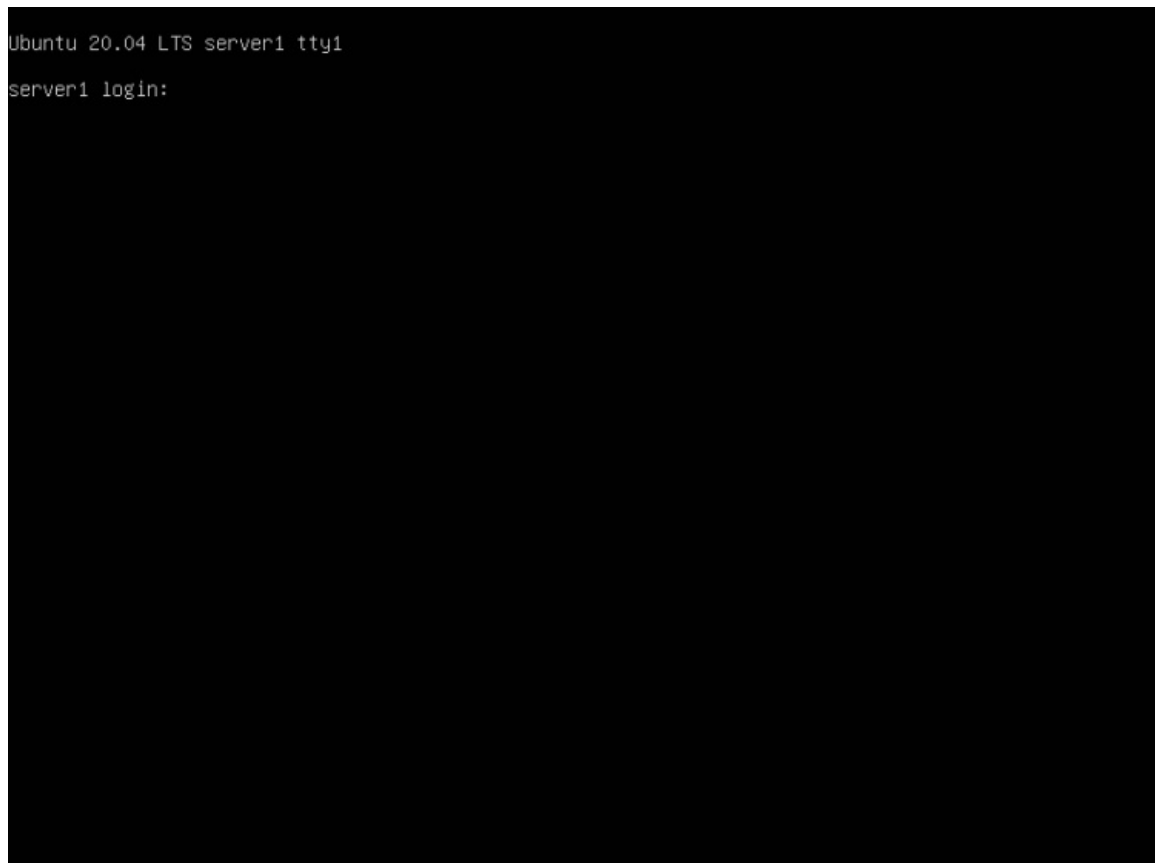
Ubuntu Installation finished successfully. Select "Reboot" to boot the server into the fresh installed Ubuntu 20.04 system.



The base installation is finished now. In the next chapter, I will explain the configuration of the static network address and install a shell based text editor for editing configuration files.

4. First Login

Now login on the shell (or remotely by SSH) on the server as user "administrator". The username may differ if you have chosen a different name during setup.



Successfully Logged into Ubuntu 20.04 Server.

```
Ubuntu 20.04 LTS server1 tty1

server1 login: administrator
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-28-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun 03 May 2020 08:24:05 AM UTC

System load:          0.48
Usage of /:           10.6% of 39.12GB
Memory usage:         10%
Swap usage:           0%
Processes:            145
Users logged in:      0
IPv4 address for ens33: 192.168.0.27
IPv6 address for ens33: 2003:e1:bf10:5300:20c:29ff:fe62:6398

0 updates can be installed immediately.
0 of these updates are security updates.

Last login: Sun May  3 08:19:28 UTC 2020 on tty1
administrator@server1:~$ _
```

5. Get root Privileges

After the reboot, you can log in with your previously created username (e.g. *administrator*). Because we must run all the steps from this tutorial with root privileges, we can either prepend all commands in this tutorial with the string *sudo*, or we become root right now by typing:

```
sudo -s
```

You can enable the root login by running:

```
sudo passwd root
```

And giving root a password. You can then directly log in as root, but this is frowned upon by the Ubuntu developers and community for various reasons. See <https://help.ubuntu.com/community/RootSudo>.)

6. Install the SSH Server (Optional)

If you did not select to install the OpenSSH server during the system installation above, you could do it now:

```
sudo apt-get -y install ssh openssh-server
```

From now on you can use an SSH client such as [PuTTY](#) and connect from your workstation to your Ubuntu 20.04 (Focal Fosse) server.

7. Install a shell-based editor (Optional)

Here we will install two text-based editors. The Nano editor is easier to use for newbies while others prefer the traditional vi/vim editor. The default *vi* program has some strange behavior on Ubuntu and Debian; to fix this, we install *vim-nox*:

```
sudo apt-get -y install nano vim-nox
```

8. Configure the Network

Because the Ubuntu installer has configured our system to get its network settings via DHCP, we can change that now because a server should have a static IP address. If you want to keep the DHCP-based network configuration, then skip this chapter. In Ubuntu 20.04, the network is configured with Netplan and the configuration file is */etc/netplan/01-netcfg.yaml*. The traditional network configuration file */etc/network/interfaces* is not used anymore. Edit

`/etc/netplan/00-installer-config.yaml` and adjust it to your needs (in this example setup I will use the IP address `192.168.0.100` and the DNS servers `8.8.4.4`, `8.8.8.8`).

Open the network configuration file with nano:

```
sudo nano /etc/netplan/00-installer-config.yaml
```

The server is using DHCP right after the install; the interfaces file will look like this:

```
# This is the network config written by 'subiquity'
network:
  ethernet:
    ens33:
      dhcp4: true
      version: 2
```

To use a static IP address `192.168.0.100`, I will change the file so that it looks like this afterward:

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernet:
    ens33:
      dhcp4: no
      dhcp6: no
      addresses: [192.168.0.100/24]
      gateway4: 192.168.0.1
      nameservers:
        addresses: [8.8.8.8,8.8.4.4]
```

IMPORTANT: The indentation of the lines matters, add the lines as shown above.

Then restart your network to apply the changes:

```
sudo netplan generate
sudo netplan apply
```

Then edit `/etc/hosts`.

```
sudo nano /etc/hosts
```

Make it look like this:

```
127.0.0.1 localhost
192.168.0.100 server1.example.com server1

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Now, we will change the hostname of our machine as follows:

```
sudo echo server1 > /etc/hostname
sudo hostname server1
```

The first command sets the hostname "server1" in the `/etc/hostname` file. This file is read by the system at boot time. The second command sets the hostname in the current session so we don't have to restart the server to apply the hostname.

As an alternative to the two commands above you can use the `hostnamectl` command which is part of the `systemd` package.

```
sudo hostnamectl set-hostname server1
```

Afterward, run:

```
hostname
hostname -f
```

The first command returns the short hostname while the second command shows the fully qualified domain name (fqdn):

```
root@server1:/home/administrator# hostname
server1
root@server1:/home/administrator# hostname -f
server1.example.com
root@server1:/home/administrator#
```

If you want to adjust the keyboard layout of the server, run this command:

```
sudo dpkg-reconfigure keyboard-configuration
```

Congratulations! Now we have a basic Ubuntu 20.04 server setup that provides a solid basis for all kinds of Ubuntu Server setups.

9. Virtual Machine image

This tutorial is available as ready to use virtual machine in OVA / OVF format for Howtoforge subscribers. The VM format is compatible with VMWare and Virtualbox and other tools that can import the ova or ovf format. You can find the download link on the right menu near the top. Click on the filename to start the download.

The login details of the VM are:

SSH Login

Username: administrator
Password: howtoforge

The administrator user has sudo permissions.

Please change the passwords after the first boot.

The VM is configured for the static IP 192.168.0.100, the IP can be changed in the file ***/etc/netplan/00-installer-config.yaml*** as shown in the tutorial step 8. The keyboard layout of the downloadable VM is the US keyboard layout.

10. Links

Ubuntu: <http://www.ubuntu.com/>

This tutorial shows the installation of an Ubuntu 20.04 (Focal Fossa) web hosting server with Apache 2.4, Postfix, Dovecot, Bind, and PureFTPd to prepare it for the installation of [ISPConfig 3.2](#). The resulting system will provide a Web, Mail, Mailinglist, DNS, and FTP Server.

ISPConfig is a web hosting control panel that allows you to configure the following services through a web browser: Apache or Nginx web server, Postfix mail server, Courier or Dovecot IMAP/POP3 server, MariaDB as MySQL replacement, BIND or MyDNS nameserver, PureFTPd, SpamAssassin, ClamAV, and many more. This setup covers the installation of Apache (instead of Nginx), BIND (instead of MyDNS), and Dovecot (instead of Courier).

1. Preliminary Note

In this tutorial, I use the hostname `server1.example.com` with the IP address `192.168.0.100` and the gateway `192.168.0.1`. These settings might differ for you, so you have to replace them where appropriate. Before proceeding further you need to have a basic minimal installation of Ubuntu 20.04 as explained in the [tutorial](#).

The commands in this tutorial have to be run with root permissions. To avoid adding `sudo` in front of each command, you'll have to become root user by running:

```
sudo -s
```

before you proceed.

2. Edit /etc/apt/sources.list and Update your Linux Installation

Edit `/etc/apt/sources.list`. Comment out or remove the installation CD from the file and make sure that the `universe` and `multiverse` repositories are enabled. It should look like this afterwards:

```
nano /etc/apt/sources.list

# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to
# newer versions of the distribution.
deb http://de.archive.ubuntu.com/ubuntu focal main restricted
# deb-src http://de.archive.ubuntu.com/ubuntu focal main restricted

## Major bug fix updates produced after the final release of the
## distribution.
deb http://de.archive.ubuntu.com/ubuntu focal-updates main restricted
# deb-src http://de.archive.ubuntu.com/ubuntu focal-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team. Also, please note that software in universe WILL NOT receive any
## review or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu focal universe
# deb-src http://de.archive.ubuntu.com/ubuntu focal universe
deb http://de.archive.ubuntu.com/ubuntu focal-updates universe
# deb-src http://de.archive.ubuntu.com/ubuntu focal-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://de.archive.ubuntu.com/ubuntu focal multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu focal multiverse
deb http://de.archive.ubuntu.com/ubuntu focal-updates multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu focal-updates multiverse

## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
deb http://de.archive.ubuntu.com/ubuntu focal-backports main restricted universe multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu focal-backports main restricted universe multiverse

## Uncomment the following two lines to add software from Canonical's
## 'partner' repository.
## This software is not part of Ubuntu, but is offered by Canonical and the
## respective vendors as a service to Ubuntu users.
# deb http://archive.canonical.com/ubuntu focal partner
# deb-src http://archive.canonical.com/ubuntu focal partner

deb http://de.archive.ubuntu.com/ubuntu focal-security main restricted
# deb-src http://de.archive.ubuntu.com/ubuntu focal-security main restricted
deb http://de.archive.ubuntu.com/ubuntu focal-security universe
# deb-src http://de.archive.ubuntu.com/ubuntu focal-security universe
deb http://de.archive.ubuntu.com/ubuntu focal-security multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu focal-security multiverse
```

Then run

```
apt-get update
```

to update the apt package database and

```
apt-get upgrade
```

to install the latest updates (if there are any). If you see that a new kernel gets installed as part of the updates, you should reboot the system afterwards:

```
reboot
```

3. Change the Default Shell

`/bin/sh` is a symlink to `/bin/dash`, however we need `/bin/bash`, not `/bin/dash`. Therefore, we do this:

```
dpkg-reconfigure dash
```

Use `dash` as the default system shell (`/bin/sh`)? <- No

If you don't do this, the ISPConfig installation will fail.

4. Disable AppArmor

AppArmor is a security extension (similar to SELinux) that should provide extended security. In my opinion, you don't need it to configure a secure system, and it usually causes more problems than advantages (think of it after you have done a week of troubleshooting because some service wasn't working as expected, and then you find out that everything was ok, only AppArmor was causing the problem). Therefore, I disable it (this is a must if you want to install ISPConfig later on).

We can disable it like this:

```
service apparmor stop
update-rc.d -f apparmor remove
apt-get remove apparmor apparmor-utils
```

5. Synchronize the System Clock

It is a good idea to synchronize the system clock with an NTP (**n**etwork **t**ime **p**rotocol) server over the Internet when you run a physical server. In case you run a virtual server then you should skip this step. Just run

```
apt-get -y install ntp
```

and your system time will always be in sync.

6. Install Postfix, Dovecot, MariaDB, rkhunter, and binutils

For installing postfix, we need to ensure that sendmail is not installed and running. To stop and remove sendmail run this command:

```
service sendmail stop; update-rc.d -f sendmail remove
```

The error message:

Failed to stop sendmail.service: Unit sendmail.service not loaded.

Is ok, it just means that sendmail was not installed, so there was nothing to be removed.

Now we can install Postfix, Dovecot, MariaDB (as MySQL replacement), rkhunter, and binutils with a single command:

```
apt-get -y install postfix postfix-mysql postfix-doc mariadb-client mariadb-server openssl getmail4 rkhunter binutils dovecot-imapd dovecot-pop3d dovecot-mysql dovecot-sieve sudo patch
```

You will be asked the following questions:

General type of mail configuration: <- Internet Site
System mail name: <- server1.example.com

It is important that you use a subdomain as "system mail name" like server1.example.com or server1.yourdomain.com and not a domain that you want to use as email domain (e.g. yourdomain.tld) later.

Next, open the TLS/SSL and submission ports in Postfix:

```
nano /etc/postfix/master.cf
```

Uncomment the submission and smtps sections as follows - add the line -o smtpd_client_restrictions=permit_sasl_authenticated,reject to both sections and leave everything thereafter commented:

```
[...]
submission inet n      -      y      -      -      smtpd
-o syslog_name=postfix/submission
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_tls_auth_only=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
smtps      inet n      -      y      -      -      smtpd
-o syslog_name=postfix/smtps
-o smtpd_tls_wrappermode=yes
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
# -o smtpd_reject_unlisted_recipient=no
# -o smtpd_client_restrictions=$mua_client_restrictions
# -o smtpd_helo_restrictions=$mua_helo_restrictions
# -o smtpd_sender_restrictions=$mua_sender_restrictions
# -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
# -o milter_macro_daemon_name=ORIGINATING
[...]
```

NOTE: The whitespaces in front of the "-o " lines are important!

Restart Postfix afterward:

```
service postfix restart
```

We want MySQL to listen on all interfaces, not just localhost. Therefore, we edit /etc/mysql/mariadb.conf.d/50-server.cnf and comment out the line bind-address = 127.0.0.1:

```
nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
[...]
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
#bind-address      = 127.0.0.1
[...]
```

Now we set a root password in MariaDB. Run:

```
mysql_secure_installation
```

You will be asked these questions:

Enter current password for root (enter for none): <- press enter
Set root password? [Y/n] <-y
New password: <- Enter the new MariaDB root password here
Re-enter new password: <- Repeat the password
Remove anonymous users? [Y/n] <-y
Disallow root login remotely? [Y/n] <-y
Reload privilege tables now? [Y/n] <-y

Set the password authentication method in MariaDB to native so we can use PHPMysqlAdmin later to connect as root user:

```
echo "update mysql.user set plugin = 'mysql_native_password' where user='root';" | mysql -u root
```

Edit the file /etc/mysql/debian.cnf and set the MYSQL / MariaDB root password there twice in the rows that start with password.

```
nano /etc/mysql/debian.cnf
```

The MySQL root password that needs to be added is shown in red. In this example, the password is "howtoforge". Replace the word "howtoforge" with the password that you have set for the MySQL root user with the mysql_secure_installation command.

```
# Automatically generated for Debian scripts. DO NOT TOUCH!
[client]
host = localhost
user = root
password = howtoforge
socket = /var/run/mysql/mysql.sock
[mysql_upgrade]
host = localhost
user = root
password = howtoforge
socket = /var/run/mysql/mysql.sock
basedir = /usr
```

Open the file /etc/security/limits.conf with an editor:

```
nano /etc/security/limits.conf
```

and add these lines at the end of the file.

```
mysql soft nfile 65535
mysql hard nfile 65535
```

Next, create a new directory /etc/systemd/system/mysql.service.d/ with the mkdir command.

```
mkdir /etc/systemd/system/mysql.service.d/
```


and add a new file inside:

```
nano /etc/systemd/system/mysql.service.d/limits.conf
```

paste the following lines into that file:

```
[Service]
LimitNOFILE=infinity
```

Save the file and close the nano editor.

Then we reload systemd and restart MariaDB:

```
systemctl daemon-reload
service mariadb restart
```

Now check that networking is enabled. Run

```
netstat -tap | grep mysql
```

The output should look like this:

```
root@server1:~# netstat -tap | grep mysql
tcp6      0      0 [::]:*          LISTEN      51836/mysql
root@server1:~#
```

7. Install Amavisd-new, SpamAssassin, and Clamav

To install amavisd-new, SpamAssassin, and ClamAV, we run

```
apt-get -y install amavisd-new spamassassin clamav clamav-daemon unzip bzip2 arj nomarch lzop cabextract apt-listchanges libnet-ldap-perl libauthen-sasl-perl clamav-docs daemon libio-string-perl libio-socket-ssl-perl libnet-ident-perl zip libnet-dns-perl postgrey
```

The ISPConfig 3 setup uses amavisd which loads the SpamAssassin filter library internally, so we can stop SpamAssassin to free up some RAM:

```
service spamassassin stop
update-rc.d -f spamassassin remove
```

To start ClamAV use:

```
freshclam
service clamav-daemon start
```

The following error can be ignored on the first run of freshclam.

```
ERROR: /var/log/clamav/freshclam.log is locked by another process
ERROR: Problem with internal logger (UpdateLogFile = /var/log/clamav/freshclam.log).
```

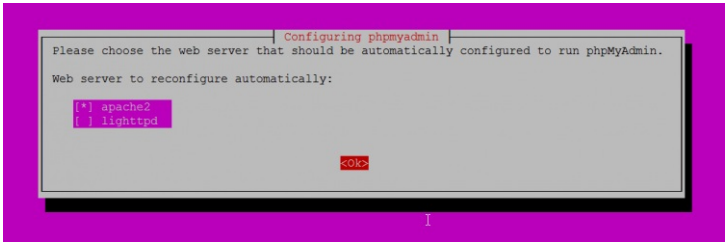
8. Install Apache, PHP, phpMyAdmin, CGI, SuExec, Pear

Apache 2.4, PHP 7.4, phpMyAdmin, CGI, suExec, and Pear can be installed as follows:

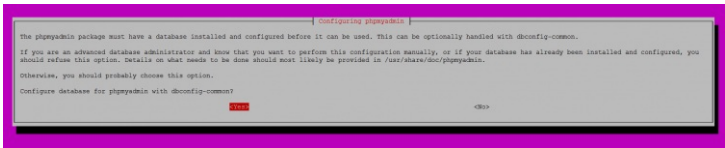
```
apt-get -y install apache2 apache2-doc apache2-utils libapache2-mod-php php7.4 php7.4-common php7.4-gd php7.4-mysql php7.4-ldap phpmyadmin php7.4-cli php7.4-cgi libapache2-mod-fcgid apache2-suexec-pristine php-pear
libruby libapache2-mod-python php7.4-curl php7.4-intl php7.4-pspell php7.4-sqlite3 php7.4-tidy php7.4-xmllrpc php7.4-xsl memcached php-memcache php-imagick php7.4-zip php7.4-mbstring php-soap php7.4-soap php7.4-
opcache php-apcu php7.4-fpm libapache2-ssl-openssl
```

You will see the following question:

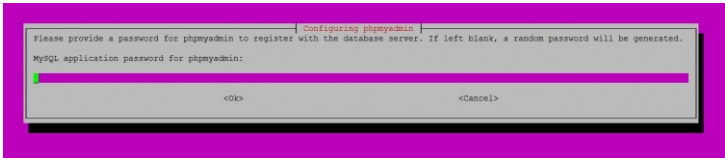
Web server to reconfigure automatically: **<- apache2**



Configure database for phpmyadmin with dbconfig-common? **<- Yes**



MySQL application password for phpmyadmin: **<- Press enter**



Then run the following command to enable the Apache modules *suexec*, *rewrite*, *ssl*, *actions*, and *include* (plus *dav*, *dav_fs*, and *auth_digest* if you want to use WebDAV):

```
a2enmod suexec rewrite ssl actions include cgi alias proxy_fcgi
```

```
a2enmod dav_fs dav auth_digest headers
```

To ensure that the server cannot be attacked through the [HTTPoxy](#) vulnerability, I will disable the HTTP_PROXY header in apache globally. Create a new httpoxy.conf file with nano:

```
nano /etc/apache2/conf-available/httpoxy.conf
```

Paste this content into the file:

```
<IfModule mod_headers.c>
    RequestHeader unset Proxy early
</IfModule>
```

Enable the config file by running:

```
a2enconf httpoxy
```

Restart Apache afterward:

```
service apache2 restart
```

If you want to host Ruby files with the extension `.rb` on your websites created through ISPConfig, you must comment out the line `application/x-ruby rb` in `/etc/mime.types`:

```
nano /etc/mime.types
```

```
[...]
#application/x-ruby          rb
[...]
```

(This is needed only for `.rb` files; Ruby files with the extension `.rbx` work out of the box.)

Restart Apache afterwards:

```
service apache2 restart
```

9. Install Let's Encrypt

ISPConfig 3.2 has built-in support for the free SSL Certificate Authority Let's encrypt. The Let's Encrypt function allows you to create free SSL Certificates for your website in ISPConfig.

Now we will add support for Let's encrypt.

```
apt-get install certbot
```

10. Install Mailman

ISPConfig allows you to manage (create/modify/delete) Mailman mailing lists. If you want to make use of this feature, install Mailman as follows:

```
apt-get -y install mailman
```

Select at least one language, e.g.:

```
Languages to support: <-- en (English)
Missing site list <-- OK
```

The error `'Job for mailman.service failed because the control process exited with error code.'` can be ignored for now.

Before we can start Mailman, a first mailing list called `mailman` must be created:

```
newlist mailman
```

```
root@server1:~# newlist mailman
Enter the email of the person running the list: <-- admin email address, e.g. listadmin@example.com
Initial mailman password: <-- admin password for the mailman list
To finish creating your mailing list, you must edit your /etc/aliases (or
equivalent) file by adding the following lines, and possibly running the
'newaliases' program:
```

```
## mailman mailing list
mailman:                "/var/lib/mailman/mail/mailman post mailman"
mailman-admin:          "/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces:        "/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm:        "/var/lib/mailman/mail/mailman confirm mailman"
mailman-join:           "/var/lib/mailman/mail/mailman join mailman"
mailman-leave:          "/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:          "/var/lib/mailman/mail/mailman owner mailman"
mailman-request:        "/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe:      "/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe:    "/var/lib/mailman/mail/mailman unsubscribe mailman"
```

Hit enter to notify mailman owner... <-- ENTER

```
root@server1:~#
```

Open `/etc/aliases` afterwards...

```
nano /etc/aliases
```

... and add the following lines:

```
[...]
## mailman mailing list
mailman:                "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin:          "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces:        "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm:        "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join:           "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave:          "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:          "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request:        "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe:      "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe:    "|/var/lib/mailman/mail/mailman unsubscribe mailman"
```

Run

```
newaliases
```

afterward and restart Postfix:

```
service postfix restart
```

Finally, we must enable the Mailman Apache configuration:

```
ln -s /etc/mailman/apache.conf /etc/apache2/conf-available/mailman.conf
```

This defines the alias `/cgi-bin/mailman/` for all Apache vhosts, which means you can access the Mailman admin interface for a list at `http://<vhost>/cgi-bin/mailman/admin/<listname>`, and the web page for users of a mailing list can be found at `http://<vhost>/cgi-bin/mailman/listinfo/<listname>`.

Under `http://<vhost>/pipemail` you can find the mailing list archives.

Activate the configuration with:

```
a2enconf mailman
```

Restart Apache afterward:

```
service apache2 restart
```


Then start the Mailman daemon:

```
service mailman start
```

11. Install PureFTPd and Quota

PureFTPd and quota can be installed with the following command:

```
apt-get -y install pure-ftpd-common pure-ftpd-mysql quota quotatool
```

Edit the file `/etc/default/pure-ftpd-common`:

```
nano /etc/default/pure-ftpd-common
```

... and make sure that the start mode is set to `standalone` and set `VIRTUALCHROOT=true`:

```
[...]
STANDALONE_OR_INETD=standalone
[...]
VIRTUALCHROOT=true
[...]
```

Now we configure PureFTPd to allow FTP and TLS sessions. FTP is a very insecure protocol because all passwords and all data are transferred in clear text. By using TLS, the whole communication can be encrypted, thus making FTP much more secure.

If you want to allow FTP and TLS sessions, run

```
echo 1 > /etc/pure-ftpd/conf/TLS
```

In order to use TLS, we must create an SSL certificate. I create it in `/etc/ssl/private/`, therefore I create that directory first:

```
mkdir -p /etc/ssl/private/
```

Afterwards, we can generate the SSL certificate as follows:

```
openssl req -x509 -nodes -days 7300 -newkey rsa:2048 -keyout /etc/ssl/private/pure-ftpd.pem -out /etc/ssl/private/pure-ftpd.pem
```

```
Country Name (2 letter code) [AU]: <-- Enter your Country Name (e.g., "DE").
State or Province Name (full name) [Some-State]:<-- Enter your State or Province Name.
Locality Name (eg, city) []:<-- Enter your City.
Organization Name (eg, company) [Internet Widgits Pty Ltd]:<-- Enter your Organization Name (e.g., the name of your company).
Organizational Unit Name (eg, section) []:<-- Enter your Organizational Unit Name (e.g., "IT Department").
Common Name (eg, YOUR name) []:<-- Enter the Fully Qualified Domain Name of the system (e.g. "server1.example.com").
Email Address []:<-- Enter your Email Address.
```

Change the permissions of the SSL certificate:

```
chmod 600 /etc/ssl/private/pure-ftpd.pem
```

Then restart PureFTPd:

```
service pure-ftpd-mysql restart
```

Edit `/etc/fstab`. Mine looks like this (I added `,usrquota=quota.user,grpquota=quota.group,jqfmt=vfsv0` to the partition with the mount point `/`):

```
nano /etc/fstab
```

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
/dev/mapper/serve1--vg-root / ext4 errors=remount-ro,usrquota=quota.user,grpquota=quota.group,jqfmt=vfsv0 0 1
/dev/mapper/serve1--vg-swap_1 none swap sw 0 0
/dev/fd0 /media/floppy0 auto rw,user,noauto,exec,utf8 0 0
```

To enable quota, run these commands:

```
mount -o remount /
```

```
quotacheck -avugm
quotaon -avug
```

Which will show the following output:

```
root@serve1:/tmp# quotacheck -avugm
quotacheck: Scanning /dev/mapper/serve1--vg-root [/] done
quotacheck: Cannot stat old user quota file //quota.user: No such file or directory. Usage will not be subtracted.
quotacheck: Cannot stat old group quota file //quota.group: No such file or directory. Usage will not be subtracted.
quotacheck: Cannot stat old user quota file //quota.user: No such file or directory. Usage will not be subtracted.
quotacheck: Cannot stat old group quota file //quota.group: No such file or directory. Usage will not be subtracted.
quotacheck: Checked 13602 directories and 96597 files
quotacheck: Old file not found.
quotacheck: Old file not found.
root@serve1:/tmp# quotaon -avug
/dev/mapper/serve1--vg-root [/]: group quotas turned on
/dev/mapper/serve1--vg-root [/]: user quotas turned on
```

12. Install BIND DNS Server

BIND can be installed as follows:

```
apt-get -y install bind9 dnstools haveged
```

Enable and start the haveged Daemon:

```
systemctl enable haveged
systemctl start haveged
```

13. Install Vlogger, Webalizer, AWStats and GoAccess

Vlogger, Webalizer, and AWStats can be installed as follows:

```
apt-get -y install vlogger webalizer awstats geoip-database libclass-dbi-mysql-perl
```

Installing the latest GoAccess version directly from GoAccess repository:

```
echo "deb https://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a /etc/apt/sources.list.d/goaccess.list
wget -O - https://deb.goaccess.io/gnugpg.key | sudo apt-key --keyring /etc/apt/trusted.gpg.d/goaccess.gpg add -
sudo apt-get update
sudo apt-get install goaccess
```

Open `/etc/cron.d/awstats` afterwards...

```
nano /etc/cron.d/awstats
```

... and comment out everything in that file:

```
#MAILTO=root

#*/10 * * * * www-data [ -x /usr/share/awstats/tools/update.sh ] && /usr/share/awstats/tools/update.sh

# Generate static reports:
#10 03 * * * www-data [ -x /usr/share/awstats/tools/buildstatic.sh ] && /usr/share/awstats/tools/buildstatic.sh
```

14. Install Jailkit

Jailkit is used to jailed shell users and cronjobs in ISPConfig. Install jailkit with apt:

```
apt-get -y install jailkit
```

15. Install fail2ban and UFW

This is optional but recommended because the ISPConfig monitor tries to show the log:

```
apt-get -y install fail2ban
```

To make fail2ban monitor PureFTPd and Dovecot, create the file `/etc/fail2ban/jail.local`:

```
nano /etc/fail2ban/jail.local
```

```
[pure-ftpd]
enabled = true
port = ftp
filter = pure-ftpd
logpath = /var/log/syslog
maxretry = 3

[dovecot]
enabled = true
filter = dovecot
action = iptables-multiport[name=dovecot-pop3imap, port="pop3,pop3s,imap,imaps", protocol=tcp]
logpath = /var/log/mail.log
maxretry = 5

[postfix]
enabled = true
port = smtp
filter = postfix
logpath = /var/log/mail.log
maxretry = 3
```

Restart fail2ban afterwards:

```
service fail2ban restart
```

To install the UFW firewall, run this apt command:

```
apt-get install ufw
```

16. Install Roundcube Webmail

To install Roundcube Webmail, run:

```
apt-get -y install roundcube roundcube-core roundcube-mysql roundcube-plugins roundcube-plugins-extra javascript-common libjs-jquery-mousewheel php-net-sieve tinymce
```

The installer might ask the following questions:

```
Configure database for roundcube with dbconfig-common? <-- Yes
MySQL application password for roundcube: <-- Press enter
```

Don't worry if you do not get these questions and a warning about the ucf script, that's ok.

The ucf warning that you will get on the shell can be ignored, it has no negative impact on the installation.

Then edit the RoundCube apache configuration file.

```
nano /etc/apache2/conf-enabled/roundcube.conf
```

and remove the `#` in front of the Alias line, then add the second Alias line for /webmail and add the line "AddType application/x-httpd-php .php" right after the "<Directory /var/lib/roundcube>" line:

```
# Those aliases do not work properly with several hosts on your apache server
# Uncomment them to use it or adapt them to your configuration
Alias /roundcube /var/lib/roundcube
Alias /webmail /var/lib/roundcube
[...]
<Directory /var/lib/roundcube>
AddType application/x-httpd-php .php
[...]
```

And restart apache

```
service apache2 restart
```

Then edit the RoundCube config.inc.php configuration file:

```
nano /etc/roundcube/config.inc.php
```

and change the default host to localhost:

```
$config['default_host'] = 'localhost';
```

and the SMTP server to:

```
$config['smtp_server'] = 'localhost';
```

and

```
$config['smtp_port'] = 25;
```

This prevents that Roundcube will show server name input field in the login form.

17. Install ISPConfig 3.2

We will use the ISPConfig 3.2 stable build here.

```
cd /tmp
wget -O ispconfig.tar.gz https://www.ispconfig.org/downloads/ISPConfig-3-stable.tar.gz
tar xzf ispconfig.tar.gz
cd ispconfig3/install/
```

The next step is to run

```
php -q install.php
```

This will start the ISPConfig 3 installer. The installer will configure all services like Postfix, Dovecot, etc. for you.

```
# php -q install.php
```

[illegible]

>> Initial configuration

Operating System: Ubuntu 20.04.1 LTS (Focal Fossa)

Following will be a few questions for primary configuration so be careful. Default values are in [brackets] and can be accepted with <ENTER>. Tap in "quit" (without the quotes) to stop the installer.

Select language (en,de) [en]: <-- Hit Enter

Installation mode (standard,expert) [standard]: <-- Hit Enter

Full qualified hostname (FQDN) of the server, eg server1.domain.tld [server1.canomi.com]: <-- Hit Enter

MySQL server hostname [localhost]: <-- Hit Enter

MySQL server port [3306]: <-- Hit Enter

MySQL root username [root]: <-- Hit Enter

MySQL root password []: <-- Enter your MySQL root password

MySQL database to create [dbispconfig]: <-- Hit Enter

MySQL charset [utf8]: <-- Hit Enter

Configuring Postgrey

Configuring Postfix

Generating a 4096 bit

.....

```
writing new private key to 'smtpd.key'
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]: <- Enter 2 letter country code
State or Province Name (full name) [Some-State]: <- Enter the name of the state
Locality Name (eg, city) []: <- Enter your city
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <- Enter company name or press enter
Organizational Unit Name (eg, section) []: <- Hit Enter
Common Name (e.g. server FQDN or YOUR name) []: <- Enter the server hostname, in my case: server1.example.com
Email Address []: <- Hit Enter

```
Configuring Mailman
Configuring Dovecot
Configuring Spamassassin
Configuring Amavisd
Configuring Getmail
Configuring BIND
Configuring Jailkit
Configuring Pureftpd
Configuring Apache
Configuring vlogger
Configuring Metronome XMPP Server
writing new private key to 'localhost.key'
```

Country Name (2 letter code) [AU]: [<- Enter 2 letter country code](#)
 Locality Name (eg, city) []: [<- Enter your city](#)
 Organization Name (eg, company) [Internet Widgits Pty Ltd]: [<- Enter company name or press enter](#)
 Organizational Unit Name (eg, section) []: [<- Hit Enter](#)
 Common Name (e.g. server FQDN or YOUR name) [server1.canomi.com]: [<- Enter the server hostname, in my case: server1.example.com](#)
 Email Address []: [<- Hit Enter](#)

```
Configuring Ubuntu Firewall
Configuring Fail2ban
[INFO] service OpenVZ not detected
Configuring Apps vhost
Installing ISPConfig
ISPConfig Port [8080]:
```

```
Admin password [admin]:
```

Do you want a secure (SSL) connection to the ISPConfig web interface (y,n) [y]: <-- Hit Enter

```
Generating RSA private key, 4096 bit long modulus
.....++
e is 65537 (0x10001)
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Country Name (2 letter code) [AU]: <- Enter 2 letter country code
State or Province Name (full name) [Some-State]: <- Enter the name of the state
Locality Name (eg, city) []: <- Enter your city
Organization Name (eg, company) [Internet Widgits Pty Ltd]: <- Enter company name or press enter
Organizational Unit Name (eg, section) []: <- Hit Enter
Common Name (e.g. server FQDN or YOUR name) []: <- Enter the server hostname, in my case: server1.example.com
Email Address []: <- Hit Enter

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: <-- Hit Enter
An optional company name []: <-- Hit Enter
writing RSA key
```

```
Symlink ISPConfig LE SSL certs to postfix? (y,n) [y]: <-- Hit Enter
```

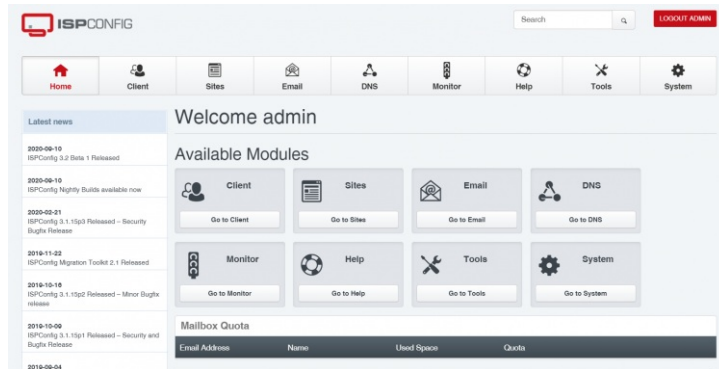
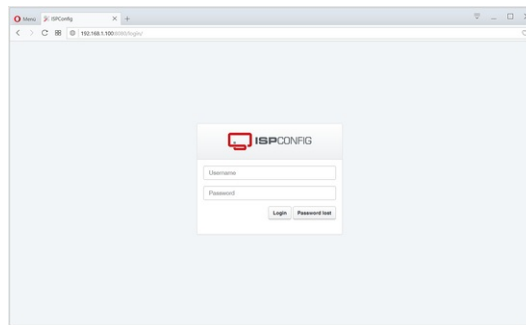
```
Symlink ISPConfig LE SSL certs to pureftpd? Creating dhparam file takes some times. (y,n) [y]: <-- Hit Enter
```

Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time

```
Configuring DBServer
Installing ISPConfig crontab
no crontab for root
no crontab for getmail
Detect IP addresses
Restarting services ...
Installation completed.
```

The installer automatically configures all underlying services, so there is no manual configuration needed.

Afterward you can access ISPConfig 3 under `http(s)://server1.example.com:8080/` or `http(s)://192.168.0.100:8080/` (HTTP or HTTPS depends on what you chose during installation). Log in with the username `admin` and the password `admin` (you should change the default password after your first login):



The system is now ready to be used.

18. Virtual machine image download of this tutorial

This tutorial is available as ready to use virtual machine image in ovf/ova format that is compatible with VMWare and Virtualbox. The virtual machine image uses the following login details:

SSH / Shell Login

Username: administrator
Password: howtoforge

This user has sudo rights.

ISPConfig Login

Username: admin
Password: howtoforge

MySQL Login

Username: root
Password: howtoforge

The IP of the VM is 192.168.0.100, it can be changed in the file `/etc/netplan/01-netcfg.yaml`. Please change all the above passwords to secure the virtual machine.

19. Links

- Ubuntu: <http://www.ubuntu.com/>
- ISPConfig: <http://www.ispconfig.org/>