

Beskrivelse:

Der skal laves en auto installation af Ubuntu Server. Det der også kaldes for en Unattended Installation. Denne opgave fokuserer på at lave en unattended installation til en Proxmox Host. Men det er ikke for svært at ændre det til hvilken som helst maskine

Forudsætninger:

Kendskab til Ubuntu/Linux.

Der bruges en del Ubuntu kommandoer i forbindelse med oprettelsen af Unattended installation mediet.

Kapitel 1: Installér Ubuntu

For at kunne oprette en Unattended ISO kræver det en "arbejds" computer. Dette kan være en virtuel, eller en fysisk PC. Det gør ingen forskel. Det første er at installere en grund ubuntu server. Husk opdatering efter installation.

Slut for Kapitel 1: Installér Ubuntu

## Kapitel 2: Forberedelse.

Log på din Ubuntu med SSH.

Vi starter med at hente en standard Ubuntu ISO, Som vi så tilretter, så den kan installere sig selv uden at vi skal blande os. Jeg bruger en ubuntu 24.04, men kan også bruges til de fleste andre versioner af ubuntu.

```
wget https://releases.ubuntu.com/24.04.1/ubuntu-24.04.1-live-server-amd64.iso
```

### Eksempel output

```
dtmek@testinst:~$ wget https://releases.ubuntu.com/24.04.1/ubuntu-24.04.1-live-server-amd64.iso
--2025-02-05 10:39:58-- https://releases.ubuntu.com/24.04.1/ubuntu-24.04.1-live-server-amd64.iso
Resolving releases.ubuntu.com (releases.ubuntu.com)... 185.125.190.37, 185.125.190.40, 91.189.91.124, ...
Connecting to releases.ubuntu.com (releases.ubuntu.com)|185.125.190.37|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2773874688 (2.6G) [application/x-iso9660-image]
Saving to: 'ubuntu-24.04.1-live-server-amd64.iso'

ubuntu-24.04.1-live-server-amd64 100%[=====>] 2.58G 63.0MB/s in 39s

2025-02-05 10:40:37 (67.0 MB/s) - 'ubuntu-24.04.1-live-server-amd64.iso' saved [2773874688/2773874688]
dtmek@testinst:~$
```

Så skal vi have lavet et underbibliotek ved navn "source-files", hvor vi placerer de filer, som vi udpakker fra den ISO fil vi lige har hentet.

```
mkdir source-files
```

### Eksempel output

```
dtmek@testinst:~$ mkdir source-files
dtmek@testinst:~$
```

Nu skal vi installere noget software, så vi kan manipulere med ISO filer (Nærmere information omkring softwaren kan findes her: [https://www.gnu.org/software/xorriso/man\\_1\\_xorriso.html](https://www.gnu.org/software/xorriso/man_1_xorriso.html)).

```
sudo apt install xorriso
```

### Eksempel output (Der kommer meget tekst, her er der kun gengivet lille del)

```
dtmek@testinst:~$ sudo apt install xorriso
[sudo] password for dtmek:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libburn4t64 libisoburnlt64 libisofs6t64
Suggested packages:
  xorriso-tcltk jigit cdck
The following NEW packages will be installed:
  libburn4t64 libisoburnlt64 libisofs6t64 xorriso
0 upgraded, 4 newly installed, 0 to remove and 5 not upgraded.
Need to get 1,081 kB of archives.
After this operation, 2,298 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://dk.archive.ubuntu.com/ubuntu noble/main amd64 libburn4t64 amd64 1.5.6-1.1build1 [158 kB]
Get:2 http://dk.archive.ubuntu.com/ubuntu noble/main amd64 libisofs6t64 amd64 1.5.6.p101-1.lubuntu2 [220 kB]
```

Nu skal vi have udpakket vores ISO fil til det underbibliotek vi kaldte "source-files". Det gør vi med det software vi har installeret lige før. Det er en lidt lang commando, jeg vil ikke gennemgå kommandoen i enkeltheder, der henvises til informationen for det software der bruges:

```
xorriso -osirrox on -indev ubuntu-24.04.1-live-server-amd64.iso --
extract_boot_images source-files/bootpart -extract / source-files
```

## Eksempel output:

```
dtmek@testinst:~$ xorriso -oslrrox on -indev ubuntu-24.04.1-live-server-amd64.iso --extract_boot_images source-files/bootpart -extract / source-files
xorriso 1.5.6 : RockRidge filesystem manipulator, libburnia project.

Copying of file objects from ISO image to disk filesystem is: Enabled
xorriso : NOTE : Loading ISO image tree from LBA 0
xorriso : UPDATE : 1065 nodes read in 1 seconds
libisofs: NOTE : Found hidden El-Torito image for EFI.
libisofs: NOTE : EFI image start and size: 1351729 * 2048 , 10144 * 512
xorriso : NOTE : Detected El-Torito boot information which currently is set to be discarded
Drive current: -indev 'ubuntu-24.04.1-live-server-amd64.iso'
Media current: stdio file, overwriteable
Media status : is written , is appendable
Boot record : El Torito , MBR protective-msdos-label grub2-mbr cyl-align-off GPT
Media summary: 1 session, 1354431 data blocks, 2645m data, 37.3g free
Volume id : 'Ubuntu-Server 24.04.1 LTS amd64'
/home/dtmek/source-files/bootpart/eltorito_catalog.img : offset=2023424 size=2048
/home/dtmek/source-files/bootpart/mbr_code_grub2.img : offset=0 size=446
/home/dtmek/source-files/bootpart/gpt_part2_efi.img : offset=2768340992 size=5193728
/home/dtmek/source-files/bootpart/systemarea.img : offset=0 size=32768
/home/dtmek/source-files/bootpart/eltorito_img1_bios.img : offset=2025472 size=32768
/home/dtmek/source-files/bootpart/eltorito_img2_uefi.img : offset=2768340992 size=5193728
xorriso : UPDATE : 720 files restored (1253.6m) in 1 seconds , 949.0xD
xorriso : UPDATE : 729 files restored (1620.4m) in 2 seconds , 277.7xD
xorriso : UPDATE : 929 files restored (1999.0m) in 3 seconds , 286.6xD
xorriso : UPDATE : 1055 files restored (2398.2m) in 4 seconds , 300.5xD
xorriso : UPDATE : 1065 files restored (2637.3m) in 5 seconds = 438.5xD
Extracted from ISO image: file '/'='/home/dtmek/source-files'
dtmek@testinst:~$
```

Slut for Kapitel 2: Forberedelse

### Kapitel 3: Rette ISO pakke til

Vi skal nu rette vores indhold fra ISO filen vi lige har udpakket, så den kan starte, og installere en Ubuntu, uden at vi skal blande os. Altså uden at indtaste informationer under installation.

Vi starter med at hoppe ned i biblioteket hvor vi har placeret vores filer fra den udpakkede ISO fil:

```
cd source-files
```

Eksempel output:

```
dtmek@testinst:~$ cd source-files
dtmek@testinst:~/source-files$ █
```

Nu skal vi lave ændringer i vores grub.cfg fil. Det er den fil der bestemmer hvad CD'en starter op med. Vi skal tilføje et par linjer for at kunne lave et menupunkt, der selv starter installationen af Ubuntu. Dette er jo Autoinstall, så vi skal ikke selv starte installationen, og indtaste data. Det skulle gerne foregå automatisk.

Vi starter med at redigere grub.cfg med vores nano editor:

```
sudo nano boot/grub/grub.cfg
```

Eksempel output på næste side.

```
set timeout=30

loadfont unicode

set menu_color_normal=white/black
set menu_color_highlight=black/light-gray

menuentry "Try or Install Ubuntu Server" {
    set gfxpayload=keep
    linux /casper/vmlinuz ---
    initrd /casper/initrd
}
grub_platform
if [ "$grub_platform" = "efi" ]; then
menuentry 'Boot from next volume' {
    exit 1
}
menuentry 'UEFI Firmware Settings' {
    fwsetup
}
else
menuentry 'Test memory' {
    linux16 /boot/memtest86+x64.bin
}
fi
```

Det første der skal ske er at "set timeout=30" sættes til værdien 3 sekunder, i stedet for 30 sekunder. Så skal vi ikke vente så længe inden installationen går i gang.

Eksempel output:

```
set timeout=3 █
```

Derefter skal vores autostart menu, der starter installationen af Ubuntu, sættes ind som det øverste punkt. Du indsætter følgende tekst lige over det øverste punkt der hedder "menuentry "Try or install Ubuntu server"" (Kopier kun de markerede linjer!).

```
##### Tekst Start #####
```

```
menuentry "Autoinstall Ubuntu Server" {  
    set gfxpayload=keep  
    linux /casper/vmlinuz quiet autoinstall  
ds=nocloud\;s=/cdrom/server/ ---  
    initrd /casper/initrd  
}
```

```
#####Tekst Slut #####
```

Efter rettelserne skal indholdet se sådan her ud:

```
set timeout=3  
  
loadfont unicode  
  
set menu_color_normal=white/black  
set menu_color_highlight=black/light-gray  
  
menuentry "Autoinstall Ubuntu Server" {  
    set gfxpayload=keep  
    linux /casper/vmlinuz quiet autoinstall ds=nocloud\;s=/cdrom/server/ ---  
    initrd /casper/initrd  
}  
  
menuentry "Try or Install Ubuntu Server" {  
    set gfxpayload=keep  
    linux /casper/vmlinuz ---  
    initrd /casper/initrd  
}  
  
grub_platform  
if [ "$grub_platform" = "efi" ]; then  
menuentry 'Boot from next volume' {  
    exit 1  
}  
menuentry 'UEFI Firmware Settings' {  
    fwsetup  
}  
else  
menuentry 'Test memory' {  
    linux16 /boot/memtest86+x64.bin  
}  
fi
```

Gem filen ved at trykke "Ctrl+x" -> trykke "Y" -> Tryk Enter ved filnavn. Og du er tilbage til normal prompt på Ubuntu.

Så skal vi lave et underbibliotek der skal hedde "server", vær sikker på at du stadig står i underbiblioteket "source-files":

```
mkdir server
```

Eksempel output:

```
dtmek@testinst:~/source-files$ mkdir server  
dtmek@testinst:~/source-files$
```

Vi skal have oprettet en tom fil under biblioteket server. Filen skal hedde "meta-data"

```
touch server/meta-data
```

Eksempel output

```
dtmek@testinst:~/source-files$ touch server/meta-data
dtmek@testinst:~/source-files$
```

Nu skal vi lave hele hjertet af operationen. Det er filen der sørger for at der kan installeres noget uden at skulle spørge om tingene. Jeg vil først gennemgå de forskellige informationer der er i filen. Den samlede fil kommer til sidst, og kan kopieres herfra, eller fra det materiale der ligger på hjemmesiden (<http://>). For nærmere information omkring hvilke funktioner der kan sættes op i denne "svar" fil, henvises til denne webside: <https://canonical-subiquity.readthedocs-hosted.com/en/latest/reference/autoinstall-reference.html>

```
#cloud-config
autoinstall:
  # version is an Autoinstall required field.
  version: 1
```

Dette skal være til stede i filen.

```
packages:
  - ubuntu-server
```

Dette fortæller at det er pakken "ubuntu-server" der skal installeres. Den kan f.eks. også være "ubuntu-workstation" der så ville installere Ubuntu workstation med grafisk brugerflade.

```
identity:
  realname: 'Peter Svendsen'
  username: pesv
  password: '$6$8oRxsavB3maVaS2i$eSMMESG6WrJLM8PKUVsUBNOaZb99PctyJz846ULOGs3eCChougC8Z
FVq5oUUBnq4VxsM.fQ7WjAerbfNxBkCy/'
  hostname: ServerNavn
```

**realname:** Er brugeres rigtige navn, her "Peter Svendsen"

**username:** Er login navnet. I det her tilfælde "pesv"

**password:** er passwordet i kryptiskform. Kan udregnes ved at køre følgende kommando i ssh:

```
openssl passwd -6 $CLEARTXT_PASSWORD
```

Eksempel output:

```
dtmek@testinst:~/source-files$ openssl passwd -6 $CLEARTXT_PASSWORD
Password:
Verifying - Password:
$6$hVStN9NmlWmHUUCH$MUCv9X8G/kwiTDlw4wSKX5gadGu4lLf207lxLRFfi75EyKVh.xgOqr4wk5wELW9yRaE48jhOWBobXOebyT2C/
dtmek@testinst:~/source-files$
```

Som man kan se genererer kommandoen den crypterede udgave af passwordet man indtastede, og man kopiere den lange streng ind på pladsen som man kan se på eksemplet i ovenstående. Her blev indtastet "Password" som password.

**Hostname:** Hvad skal server hedde.

Denne del indeholder alt omkring oprettelse af brugere, og password til samme.

```
locale: en_UK.UTF-8
keyboard:
  layout: dk
```

**locale:** Hvilken codepage skal bruges (det med hvilket tegnsæt osv.)

**Keyboard layout:** Hvilken layout er keyboard, her er det dk (Dansk)

Hvordan skal tastatur, og codepage sætte op.

```
timezone: "Europe/Copenhagen"
```

Hvilken tidszone kører computeren.

```
package_update: true
package_upgrade: true
```

Kør både en apt update, og apt upgrade efter installation.

```
storage:
  swap:
    size: 0
  layout:
    name: direct
```

Opsætning af harddisk partitionering, "**name: direct**" betyder at den ikke får LVM sat op.

```
ssh:
  allow-pw: true
  install-server: true
  authorized-keys:
    - ssh-key 1
    - ssh-key 2
```

Installer SSH server, og konfigurer samme.

```
network:
  network:
    version: 2
  ethernets:
    ens18:
      dhcp4: true
      dhcp-identifier: mac
```

Dette installerer netværk. Same format som netplan opsætning.

Hele **network:** elementet kan fjernes, og så tager den automatisk dhcp configuration, hvis den har driver til netkort. Dette kræver dog at der er et netværk til rådighed på det netkort der skal installeres, og at der på det netværk er adgang til en DHCP server.

```

late-commands:
- curtin in-target -- update-grub
- curtin in-target -- apt-get install -y cloud-init
- curtin in-target -- apt-get autoremove -y
# Kun hvis maskine skal køre på Proxmox
- curtin in-target -- apt-get install -y qemu-guest-agent
- curtin in-target -- apt-get update
- curtin in-target -- apt-get upgrade -y
- curtin in-target -- apt-get dist-upgrade -y

```

Dette er kommandoer der kører efter maskinen er blevet installeret. Der anbefales at køre de første 3 kommandoer (De kommandoer der er indtil "# Kun hvis maskine kører Proxmox") Man kan selv tilføje ekstra kommandoer. Her har jeg tilføjet installationen af klient driver til Proxmox clienter, og jeg kører en update, upgrade osv.

For at oprette filen, bruger vi editoren nano:

```
nano server/user-data
```

Filen skal indeholde følgende (Kopier teksten herunder, og sæt det ind i dit SSH vindue ved at højreklikke i vinduet):

```

##### Filen user-data #####

#cloud-config
autoinstall:
  # version is an Autoinstall required field.
  version: 1

  # This adds the default ubuntu-desktop packages to the system.
  # Any desired additional packages may also be listed here.
  packages:
    - ubuntu-server

  # User creation can occur in one of 3 ways:
  # 1. Create a user using this `identity` section.
  # 2. Create users as documented in cloud-init inside the user-data
section,
  # which means this single-user identity section may be removed.
  # 3. Prompt for user configuration on first boot. Remove this identity
  # section and see the "Installation without a default user" section.
  identity:
    realname: 'dtmek'
    username: dtmek
    # If A password hash is needed. `openssl passwd -6
$CLEARTEXT_PASSWORD` can help.
    password: ''
    hostname: ServerNavn

  locale: en_UK.UTF-8
  keyboard:
    layout: dk

  # Seet timezone for Copenhagen
  timezone: "Europe/Copenhagen"

```



```

package_update: true
package_upgrade: true

# Subiquity will, by default, configure a partition layout using LVM.
# The 'direct' layout method shown here will produce a non-LVM result.
storage:
  swap:
    size: 0
  layout:
    name: direct

ssh:
  allow-pw: true
  install-server: true
  authorized-keys:
    - ssh-key 1
    - ssh-key 2

# All of network can be removed, and the installation will create the
network automatically, if it has access to the driver.
network:
  network:
    version: 2
  ethernet:
    ens18:
      dhcp4: true
      dhcp-identifier: mac

late-commands:
  - curtin in-target -- update-grub
  - curtin in-target -- apt-get install -y cloud-init
  - curtin in-target -- apt-get autoremove -y
# Kun hvis maskine skal køre på Proxmox
  - curtin in-target -- apt-get install -y qemu-guest-agent
  - curtin in-target -- apt-get update
  - curtin in-target -- apt-get upgrade -y
  - curtin in-target -- apt-get dist-upgrade -y

##### Slut på Filen user-data #####

```

Gem filen ved at trykke "Ctrl+x" -> trykke "Y" -> Tryk Enter ved filnavn. Og du er tilbage til normal prompt på Ubuntu.

Slut for Kapitel 3: Rette ISO pakke til

## Kapitel 4: Generer den nye ISO fil

Nu har vi lavet alle ændringer til de filer vi skal bruge til at lave den nye ISO fil. Vi skal også kunne skrive vores ændringer om til en ny ISO fil. Her bruger vi igen xorriso (Dette er en MEGET lang kommando, og ja der er et punktum til sidst):

```
xorriso -as mkisofs -r -V "ubuntu-autoinstall" -J -boot-load-size 4 -  
boot-info-table -input-charset utf-8 -eltorito-alt-boot -b  
bootpart/eltorito_img1_bios.img -no-emul-boot -o ../installer.iso .
```

### Eksempel output:

```
dtmek@testinst:~/source-files$ xorriso -as mkisofs -r -V "ubuntu-autoinstall" -J -boot-load-size 4 -boot-info-table -input-charset utf-8 -eltorito-alt-boot -b bootpart/eltorito_img1_bios.img -no-emul-boot -o ../installer.iso .  
xorriso 1.5.6 : RockRidge filesystem manipulator, libburnia project.  
  
xorriso : NOTE : Local character set is now assumed as: 'utf-8'  
Drive current: -outdev 'stdio:../installer.iso'  
Media current: stdio file, overwriteable  
Media status : is Blank  
Media summary: 0 sessions, 0 data blocks, 0 data, 34.7g free  
xorriso : WARNING : -wolid text does not comply to ISO 9660 / ECMA 119 rules  
Added to ISO image: directory '/"/"/home/dtmek/source-files'  
xorriso : UPDATE : 1075 files added in 1 seconds  
xorriso : UPDATE : 1075 files added in 1 seconds  
libisofs: WARNING : Cannot add /distro/stable to Joliet tree. Symlinks can only be added to a Rock Ridge tree.  
libisofs: WARNING : More symbolic links were omitted from Joliet tree.  
libisofs: WARNING : Number of symbolic links omitted from Joliet tree: 3  
xorriso : UPDATE : 4.38% done  
xorriso : UPDATE : 81.44% done  
xorriso : UPDATE : 65.53% done, estimate finish Wed Feb 05 12:17:39 2025  
xorriso : UPDATE : 83.49% done  
ISO image produced: 1356300 sectors  
Written to medium : 1356300 sectors at LBA 0  
Writing to 'stdio:../installer.iso' completed successfully.  
dtmek@testinst:~/source-files$
```

For at se filen den har genereret, hopper vi en tand tilbage:

```
cd ..
```

### Eksempel output:


```
dtmek@testinst:~/source-files$ cd ..  
dtmek@testinst:~$
```

Og vi lister hvad der er i biblioteket:

```
ls -l
```

### Eksempel output:

```
dtmek@testinst:~$ ls -l  
total 5421876  
-rw-rw-r-- 1 dtmek dtmek 2778112000 Feb  5 12:17 installer.iso  
drwxrwxr-x 11 dtmek dtmek      4096 Feb  5 11:12 source-files  
-rw-rw-r-- 1 dtmek dtmek 2773874688 Aug 27 15:40 ubuntu-24.04.1-live-server-amd64.iso  
dtmek@testinst:~$
```



Vores nye ISO fil er ved pilen, og hedder installer.iso. Så den har generet en ISO fil med de ændringer vi har lavet.

Derefter kan du boote fra den genererede ISO fil, og den maskine du booter fra, burde starte op og selv installere Ubuntu server. Uden der skal indtastes nogen informationer

Slut for Kapitel 4: Generer den nye ISO fil

Dette er så også slutningen på denne opgave.