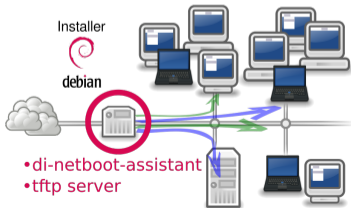


# Debian automatisch ausrollen und konfigurieren

PXE, di-netboot-assistant, preseeding, ansible

Andreas B. Mundt  
andi@debian.org

Chemnitzer Linuxtage, 15.3.2020



Debian ausrollen  
und konfigurieren

Andreas B. Mundt

PXE-Installation

Netzwerk-Installation  
di-netboot-assistant

Automatisierung

„Preseeding“

Beispiel: Temporärer  
Paket-Cache

Beispiel: „preseed“-Datei  
über TFTP laden

Konfigurations-  
Management mit  
Ansible

debian-installer/preseeding  
und ansible kombiniert

Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele  
installbox

kiosk und cloudbox

kerberos und  
kerberos-client

Zusammenfassung

Anhang

PXE und TFTP

di-netboot-assistant

Workshop

# Ausblick

TFTP/PXE-Boot-Installation mit di-netboot-assistant

Automatisierung der Installation

Ansible-Playbooks

Zusammenfassung



Debian ausrollen  
und konfigurieren

Andreas B. Mundt

## PXE-Installation

Netzwerk-Installation  
di-netboot-assistant

## Automatisierung

„Preseeding“

Beispiel: Temporärer  
Paket-Cache

Beispiel: „preseed“-Datei  
über TFTP laden

Konfigurations-  
Management mit  
Ansible

debian-installer/preseeding  
und ansible kombiniert

## Ansible-Playbooks

Das Debian-LAN Projekt  
Ansible-Playbook Beispiele  
installbox

kiosk und cloudbox  
kerberos und  
kerberos-client

## Zusammenfassung

## Anhang

PXE und TFTP  
di-netboot-assistant  
Workshop

# Teil 1: Installation übers Netzwerk

## TFTP/PXE-Boot-Installation mit di-netboot-assistant Netzwerk-Installation di-netboot-assistant

## Automatisierung der Installation

## Ansible-Playbooks

## Zusammenfassung



# Wann und warum? Vor- und Nachteile

Wann?  $\Rightarrow$  Mehrere Clients sollen (evtl. mehrfach) installiert werden.

Vorteile:

- ▶ Unterschiedliche Installationsmedien (Distributionen, Architekturen, ...)
- ▶ Zentrale Bereitstellung und Aktualisierung
- ▶ Keine zusätzliche Hardware (USB-Stick, CD) erforderlich
- ▶ Parallele Installation
- ▶ Einfache und flexible Anpassung der Installation
- ▶ Weitgehende Hardware- (bzw. Firmware-) Unterstützung

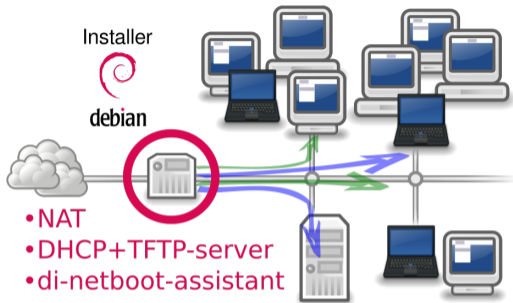
Nachteile: Zusätzliche Infrastruktur (bzw. administrativen Zugriff darauf):

- ▶ LAN/Subnet (evtl. virtuell)
  - ▶ DHCP-Server
  - ▶ TFTP-Server

u.a.  $\rightarrow$  dnsmasq



# Wie? TFTP/PXE-Boot-Installation



- ▶ Client → PXE-boot, Client fordert über DHCP IP-Adresse und Infos an
- ▶ DHCP-Server → Info über TFTP Server und *Network Bootstrap Program*
- ▶ Client lädt `pxelinux.0` (legacy BIOS) bzw. `bootnetx64.efi` (UEFI) vom TFTP-Server
- ▶ Client lädt Kernel, Initramfs und ... vom TFTP-server ...

⇒ Installation kann starten!



# di-netboot-assistant

## Das Debian Paket „di-netboot-assistant“

Vereinfacht die Einrichtung des DHCP/TFTP-Servers:

- ▶ Lädt erforderliche Dateien herunter, prüft Signaturen, extrahiert und organisiert sie
- ▶ Unterstützung für PXElinux, grub-efi und u-boot
- ▶ Auswahlmenu
- ▶ „non-free“-Firmware
- ▶ Automatisierung mittels „preseeding“
- ▶ Konfigurationsmanagement mittels ansible/ansible-pull



# Welche Installer-Images sind verfügbar?

Paket installieren:

```
apt install di-netboot-assistant # dnsmasq --no-install-recommends
```

Verfügbare Images:

```
di-netboot-assistant install
```

```
E: No repository name was passed for 'install'.
```

```
I: Declared repositories are:
```

```
bullseye bullseye-gtk buster buster-gtk daily daily-gtk  
jessie jessie-gtk oldstable oldstable-gtk precise stable  
stable-gtk stretch stretch-gtk testing testing-gtk trusty  
vivid wheezy wheezy-gtk wily xenial yakkety zesty
```



# Installer-Images und Firmware einrichten

```
di-netboot-assistant install stable testing
```

```
I: Processing stable/amd64.
```

```
I: Downloading 'SHA256SUMS'.
```

```
I: Good signature from "Debian Archive Automatic Signing Key
```

```
[...]
```

```
I: Building menu entries for the netboot-images.
```

```
I: • stable-amd64
```

```
I: • testing-amd64
```

```
[...]
```

```
di-netboot-assistant fw-toggle stable
```

```
I: Processing non-free firmware for stable/amd64.
```

```
I: Downloading 'SHA256SUMS'.
```

```
[...]
```

```
I: Downloading 'firmware.cpio.gz'.
```

```
I: Adding non-free firmware.
```



Dann den Client über PXE booten ...

## PXE-Installation

Netzwerk-Installation

di-netboot-assistant

## Automatisierung

„Preseeding“

Beispiel: Temporärer  
Paket-Cache

Beispiel: „preseed“-Datei  
über TFTP laden

Konfigurations-  
Management mit  
Ansible

debian-installer/preseeding  
und ansible kombiniert

## Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele  
installbox

kiosk und cloudbox  
kerberos und  
kerberos-client

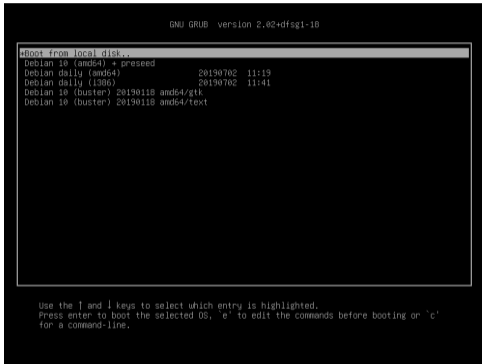
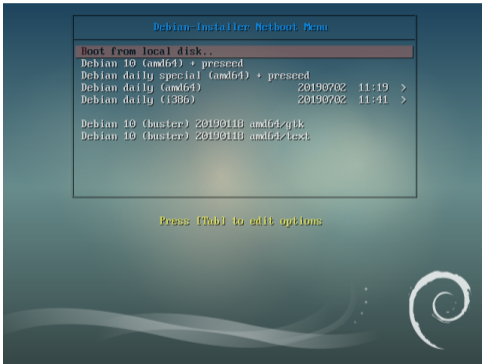
## Zusammenfassung

## Anhang

PXE und TFTP

di-netboot-assistant

Workshop



Andreas B. Mundt

## PXE-Installation

Netzwerk-Installation

dl-netboot-assistant

## Automatisierung

„Preseeding“

Beispiel: Temporärer Paket-Cache

Beispiel: „preseed“-Datei über TFTP laden

Konfigurations-Management mit Ansible

debian-installer/preseeding und ansible kombiniert

## Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele  
installbox

kiosk und cloudbox

kerberos und kerberos-client

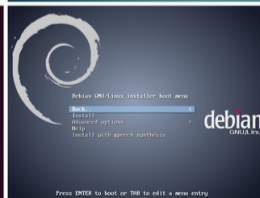
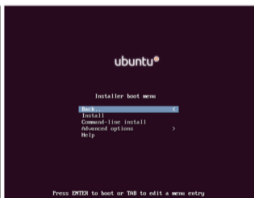
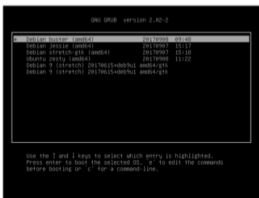
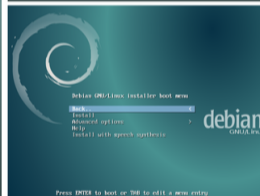
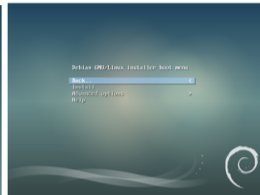
## Zusammenfassung

## Anhang

PXE und TFTP

dl-netboot-assistant

Workshop



## Vorführung di-netboot-assistant

```
sudo apt --no-install-recommends install di-netboot-assistant
sudo di-netboot-assistant install buster
TFTP_ROOT="/var/lib/tftpboot/"
qemu-system-x86_64 -enable-kvm -m 1024 -boot n -net nic -net user,tftp=$TFTP_ROOT,bootfile=/d-i/n-a/pxelinux.0
```



# Teil 2: Automatisierung

## TFTP/PXE-Boot-Installation mit di-netboot-assistant

### Automatisierung der Installation

#### „Preseeding“

Beispiel: Temporärer Paket-Cache

Beispiel: „preseed“-Datei über TFTP laden

Konfigurations-Management mit Ansible

debian-installer/preseeding und ansible kombiniert

### Ansible-Playbooks

### Zusammenfassung



# „Preseeding“: Antwort auf (alle) Fragen

„preseeding“ = Antworten für während der Installation auftretenden Fragen bereitstellen

⇒ Vorkonfigurations-Datei / „preseed“-Datei

Methoden:

- ▶ Initial RAM Disk (initrd)
- ▶ Datei auf USB-Stick
- ▶ Über das Netzwerk:
  - ▶ HTTP
  - ▶ TFTP
- ▶ Boot Parameter



# Beispiel: Boot-Parameter mit di-netboot-assistant

Angepasster Boot-Menu-Eintrag in

`/etc/di-netboot-assistant/{pxelinux.HEAD,grub.cfg.HEAD}:`

```
/etc/di-netboot-assistant/pxelinux.HEAD
```

```
## Use temporary package cache during installation, other default locale,  
##                                     install etckeeper.
```

```
LABEL tmp pkg cache
```

```
    MENU LABEL Debian stable (amd64) + temporary package cache + locale
```

```
    kernel ::/d-i/n-a/stable/amd64/linux
```

```
    append initrd=::/d-i/n-a/stable/amd64/initrd.gz \
```

```
        mirror/http/proxy?=http://192.168.122.1:3142/ \
```

```
        locale?=de_DE pkgsel/include=etckeeper \
```

```
        preseed/late_command="rm -fv /target/etc/apt/apt.conf" ---
```

```
##
```

⇒ bei kleinen Anpassungen, einzelnen Fragen



## Bereitstellung der „preseed“-Datei

- ▶ Beispiel „preseed“-Datei<sup>1</sup> unter:  
`/usr/share/doc/di-netboot-assistant/examples/preseed.cfg`
- ▶ Verwende bereits vorhandenen TFTP server:

```
$TFTP_ROOT/d-i/buster/:
```

```
# mkdir -p /var/lib/tftpboot/d-i/buster/  
# cp /usr/share/doc/di-netboot-assistant/examples/preseed.cfg \  
    /var/lib/tftpboot/d-i/buster/
```

- ▶ Anpassungen:
  - ▶ Sprache und Tastatur
  - ▶ Partitionierung
  - ▶ ssh public key
- ▶ TFTP-Server Name muss auflösbar sein, andernfalls IP-Adresse verwenden

<sup>1</sup>Siehe auch: <https://www.debian.org/releases/stable/example-preseed.txt>



## Beispiel: „preseed“-Datei über TFTP laden

Entsprechend angepasster Boot-Menu-Eintrag in

```
/etc/di-netboot-assistant/{pxelinux.HEAD,grub.cfg.HEAD}:
```

```
/etc/di-netboot-assistant/pxelinux.HEAD
```

```
## Install stable automatically.
```

```
LABEL autoinstall
```

```
MENU LABEL Debian stable (amd64) + preseed
```

```
kernel ::/d-i/n-a/stable/amd64/linux
```

```
append initrd=::/d-i/n-a/stable/amd64/initrd.gz \
```

```
auto=true priority=critical url=tftp://installbox ---
```

```
##
```

Die „preseed“-Datei `d-i/buster/preseed.cfg` wird vom TFTP-Server  
„installbox“ geladen.





# „Preseeding“ und modernes Konfigurations-Management

„Preseeding“ ist mit modernem Konfigurations-Management nicht vergleichbar.

Darum:

- ▶ „preseeding“ nur für das Nötigste
- ▶ Weitere Konfiguration mit einem modernen Konfigurations-Management-System

Vorteile:

- ▶ „Preseeding“ einmal für alle Systeme
- ▶ Alle weiteren Vorteile des Konfigurations-Management-Systems wie Gruppen, Tags, Rollen, Logging, . . . .

Im Weiteren als Konfigurations-Management-System: **Ansible**



# Elemente des Konfigurations-Managements mit Ansible

Man hat:

- ▶ Rechner (host) und Gruppen von Rechnern (group of hosts)
- ▶ Rollen (role), bestehend u.a. aus
  - ▶ Aufgaben (tasks)
  - ▶ Dateien (files)
  - ▶ Vorlagen (templates)
  - ▶ Reaktionen (handlers)
- ▶ playbooks (Kombination von allem)



# Ausrollen einer Konfiguration mit Ansible: ansible-playbook

Entweder: ansible-playbook

- ▶ Erforderlich: Private/Public Key SSH Login für den Rechner als root oder sudo user
- ▶ Das „playbook“ (configuration space) wird auf dem Client implementiert (push config)

Vorbereitung in der „preseed“-Datei:

```
### This command is run just before the install finishes:
```

```
d-i preseed/late_command string \  
    mkdir -p /target/home/ansible/.ssh && \  
    echo "ssh-rsa YOURSSHPUBLICKEYHERE ansible@installbox" \  
        >> /target/home/ansible/.ssh/authorized_keys ; \  
in-target chown -R ansible:ansible /home/ansible/.ssh/ ; \  
in-target chmod -R og= /home/ansible/.ssh/
```

Nachteil: Zweiteilige Installation bis zum finalen Systemzustand  
Vorteil: Keine Probleme durch chroot-Umgebung des Installers



# Ausrollen einer Konfiguration mit Ansible: ansible-pull

Alternativ: ansible-pull

- ▶ Der Client lädt „playbooks“ (configuration space) aus dem Netz<sup>2</sup>
- ▶ und implementiert das relevante playbook (pull config)

Vorbereitung in der „preseed“-Datei:

```
### Individual additional packages to install
d-i pkgsel/include string firmware-linux ansible git
[...]
### This command is run just before the install finishes:
d-i preseed/late_command string \
    mkdir -p /target/home/ansible/.ssh && \
    echo "ssh-rsa YOURSSHPUBLICKEYHERE ansible@installbox" \
        >> /target/home/ansible/.ssh/authorized_keys ; \
    in-target chown -R ansible:ansible /home/ansible/.ssh/ ; \
    in-target chmod -R og= /home/ansible/.ssh/ ; \
    in-target ansible-pull --verbose --purge \
    -i localhost, --url=git://installbox/.git $playbook
```

<sup>2</sup> git daemon -verbose -export-all -base-path=/dir/of/playbook - /dir/of/playbook

# Angepasster Boot-Menu-Eintrag in {pxelinux.HEAD, grub.cfg.HEAD}:

Angepasster Boot-Menu-Eintrag in {pxelinux.HEAD, grub.cfg.HEAD}:

```
/etc/di-netboot-assistant/pxelinux.HEAD
```

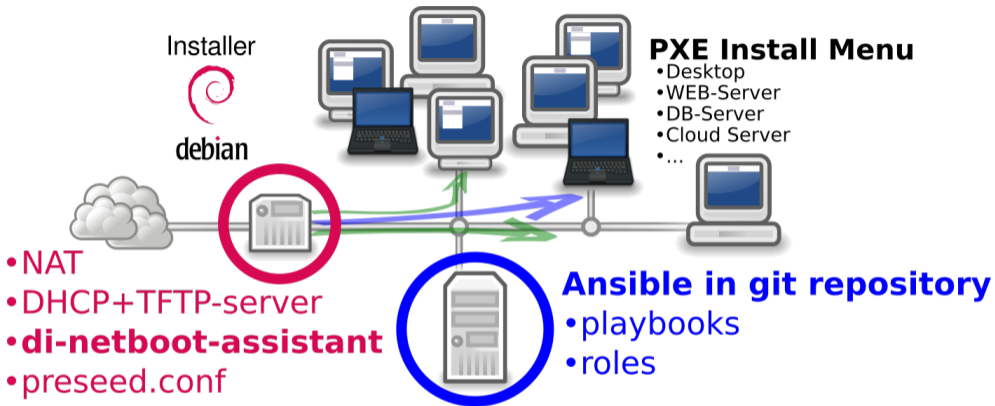
```
MENU LABEL Debian stable (amd64) + preseed + WEBSERVER
kernel ::/d-i/n-a/stable/amd64/linux
append initrd=::/d-i/n-a/stable/amd64/initrd.gz \
        auto=true priority=critical url=tftp://installbox \
        playbook=webserver.yml ---
```

```
MENU LABEL Debian testing (amd64) + preseed + DESKTOP
kernel ::/d-i/n-a/stable/amd64/linux
append initrd=::/d-i/n-a/stable/amd64/initrd.gz \
        auto=true priority=critical url=tftp://installbox \
        playbook=desktop.yml ---
```

Vorteil: Installation bis zum finalen Systemzustand in einem Zug  
Nachteil: Mögliche Probleme durch chroot-Umgebung des Installers



# Nochmal alles zusammen:



Debian ausrollen  
und konfigurieren

Andreas B. Mundt

PXE-Installation

Netzwerk-Installation  
di-netboot-assistant

Automatisierung

„Preseeding“

Beispiel: Temporärer  
Paket-Cache

Beispiel: „preseed“-Datei  
über TFTP laden

Konfigurations-  
Management mit  
Ansible

debian-installer/preseeding  
und ansible kombiniert

Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele  
installbox

kiosk und cloudbox

kerberox und  
kerberox-client

Zusammenfassung

Anhang

PXE und TFTP

di-netboot-assistant

Workshop

# Teil 3: Ansible-Playbooks

TFTP/PXE-Boot-Installation mit `di-netboot-assistant`

Automatisierung der Installation

## Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele

`installbox`

`kiosk` und `cloudbox`

`kerberox` und `kerberox-client`

Zusammenfassung



# Idee: Das Debian-LAN-Projekt

- ▶ Der Debian-Installer ermöglichen einzelne Rechner einfach zu installieren.
- ▶ Das Debian-LAN-Projekt<sup>3</sup> versucht, das Ausrollen einer ganzen Netzwerkumgebung zu unterstützen.

Vor diesem Hintergrund stehen die im Folgenden vorgestellten  
ansible-Playbooks:

```
https://salsa.debian.org/andi/debian-lan-ansible
```

```
git clone https://salsa.debian.org/andi/debian-lan-ansible.git
```

Beiträge jeglicher Art, wie Patches, Pull-Requests und/oder weitere Rollen sind  
dazu jederzeit höchst willkommen!



---

<sup>3</sup><https://wiki.debian.org/DebianLAN>



## Playbook Beispiel: `installbox.yml`

- ▶ `installbox.yml`: Ausrollen weiterer automatischer Installationen im LAN
  - ▶ 2 NICs, zu externem WAN und „eigenem“ LAN
  - ▶ TFTP Installationen im LAN
  - ▶ „Preseeding“ und `ansible`-Playbooks
  - ▶ Paket-Cache



Laptop als Gateway zwischen WAN (über WLAN) und LAN (über Kabel)



# Playbook Beispiele: kiosk.yml, cloudbox.yml

- ▶ kiosk.yml: Schulrechner, Hackerspace, ...
  - ▶ Automatische Anmeldung
  - ▶ Benutzerfreundliche Standardeinstellungen
  - ▶ Temporäres Home-Verzeichnis (tmpfs), wird beim Booten gelöscht
- ▶ cloudbox.yml: Nextcloud-Server
  - ▶ Nextcloud automatisch ausrollen
  - ▶ Dynamischer DNS Name
  - ▶ Backup mit borg
  - ▶ ...



# Playbook Beispiele: kerberos.yml, kerberos-client.yml

- ▶ kerberos.yml: Server für minimalistisches, kerberisiertes LAN
  - ▶ Automatische Installation der Client Rechner
  - ▶ Zentrale Home-Verzeichnisse auf dem Kerberos-Server (kerberisiertes NFSv4)
  - ▶ Einfaches Skript zur Benutzer- und Maschinenverwaltung (KDC-LDAP)
- ▶ kerberos-client.yml: Angepasster Client für das kerberisierte LAN

## ToDo/WIP:

- ▶ Einbindung der Nextcloud-Rolle (LDAP, SSO)
- ▶ XMPP Server
- ▶ ...

⇒ Komplette LAN Umgebung



Debian ausrollen  
und konfigurieren

Andreas B. Mundt

PXE-Installation

Netzwerk-Installation  
di-netboot-assistant

Automatisierung

„Preseeding“  
Beispiel: Temporärer  
Paket-Cache  
Beispiel: „preseed“-Datei  
über TFTP laden  
Konfigurations-  
Management mit  
Ansible  
debian-installer/preseeding  
und ansible kombiniert

Ansible-Playbooks

Das Debian-LAN Projekt  
Ansible-Playbook Beispiele  
installbox  
kiosk und cloudbox  
kerberos und  
kerberos-client

Zusammenfassung

Anhang

PXE und TFTP  
di-netboot-assistant  
Workshop

# Zusammenfassung und Fragen

TFTP/PXE-Boot-Installation mit di-netboot-assistant

Netzwerk-Installation

di-netboot-assistant

Automatisierung der Installation

„Preseeding“

Konfigurations-Management mit Ansible

debian-installer/preseeding und ansible kombiniert

Ansible-Playbooks

Das Debian-LAN Projekt

Ansible-Playbook Beispiele

Zusammenfassung

Vielen Dank für die Aufmerksamkeit!



## PXE-Installation

Netzwerk-Installation  
di-netboot-assistant

## Automatisierung

„Preseeding“  
Beispiel: Temporärer  
Paket-Cache  
Beispiel: „preseed“-Datei  
über TFTP laden  
Konfigurations-  
Management mit  
Ansible  
debian-installer/preseeding  
und ansible kombiniert

## Ansible-Playbooks

Das Debian-LAN Projekt  
Ansible-Playbook Beispiele  
installbox  
kiosk und cloudbox  
kerberos und  
kerberos-client

## Zusammenfassung

## Anhang

PXE und TFTP  
di-netboot-assistant  
Workshop

# Weitere Links and Quellen

- ▶ **di-netboot-assistant package:**  
<https://packages.debian.org/di-netboot-assistant>  
<https://salsa.debian.org/installer-team/netboot-assistant>
- ▶ **Debian Documentation “Preseeding”:**  
<https://www.debian.org/releases/stretch/amd64/apb.html.en>
- ▶ **Debian Wiki:**  
<https://wiki.debian.org/DebianInstaller/NetbootAssistant>  
<https://wiki.debian.org/DebianLAN>
- ▶ **Debian-LAN presentations:**  
<https://people.debian.org/~andi/>
- ▶ **Debian-LAN-Ansible:**  
<https://salsa.debian.org/andi/debian-lan-ansible>

Illustrations remixed from: <https://openclipart.org/>



# PXE boot: legacy BIOS

Client:

```
Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+git-20190125.36a4c85-1 -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP iSCSI NFS TFTP AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:9e:e7:25 using virtio-net on 0000:00:03.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:9e:e7:25)..... ok
net0: 192.168.0.129/255.255.255.0 gw 192.168.0.10
net0: fe80::5054:ff:fe9e:e725/64
Next server: 192.168.0.10
Filename: d-i/n-a/pxelinux.0
tftp://192.168.0.10/d-i/n-a/pxelinux.0... ok
pxelinux.0 : 42425 bytes [PXE-NBP]

PXELINUX 6.04 PXE 20181203 Copyright (C) 1994-2015 H. Peter Anvin et al
```

## PXE-Installation

Netzwerk-Installation  
d1-netboot-assistant

## Automatisierung

„Preseeding“  
Beispiel: Temporärer  
Paket-Cache  
Beispiel: „preseed“-Datei  
über TFTP laden

Konfigurations-  
Management mit  
Ansible  
debian-installer/preseeding  
und ansible kombiniert

## Ansible-Playbooks

Das Debian-LAN Projekt  
Ansible-Playbook Beispiele  
installbox  
kiosk und cloudbox  
kerberos und  
kerberos-client

## Zusammenfassung

## Anhang

PXE und TFTP  
d1-netboot-assistant  
Workshop



# PXE boot: legacy BIOS

## DHCP/TFTP Server:

```
$ journald -f
```

```
dnsmasq-dhcp: DHCPDISCOVER(enp2s0) 52:54:00:9e:e7:25
```

```
dnsmasq-dhcp: DHCPOFFER(enp2s0) 192.168.0.129 52:54:00:9e:e7:25
```

```
dnsmasq-dhcp: DHCPDISCOVER(enp2s0) 52:54:00:9e:e7:25
```

```
dnsmasq-dhcp: DHCPOFFER(enp2s0) 192.168.0.129 52:54:00:9e:e7:25
```

```
dnsmasq-dhcp: DHCPREQUEST(enp2s0) 192.168.0.129 52:54:00:9e:e7:25
```

```
dnsmasq-dhcp: DHCPACK(enp2s0) 192.168.0.129 52:54:00:9e:e7:25 debian079
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/pxelinux.0 to 192.168.0.129
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/ldlinux.c32 to 192.168.0.129
```

```
dnsmasq-tftp: file /var/lib/tftpboot/d-i/n-a/pxelinux.cfg/COA80081 not found
```

```
[...]
```

```
dnsmasq-tftp: file /var/lib/tftpboot/d-i/n-a/pxelinux.cfg/C0 not found
```

```
dnsmasq-tftp: file /var/lib/tftpboot/d-i/n-a/pxelinux.cfg/C not found
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/pxelinux.cfg/default to 192.168.0.129
```



# PXE boot: UEFI

Client:

```
iPXE 1.0.0+git-20190125.36a4c85-1 -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP iSCSI NFS TFTP AoE EFI Menu

net0: 52:54:00:e9:a4:74 using virtio-net on 0000:01:00.0 (open)
  [Link:up, TX:0 TXE:0 RX:6 RXE:1]
  [RXE: 1 x "Operation not supported (http://ipxe.org/3c3f6383)"]
Configuring (net0 52:54:00:e9:a4:74) ..... ok
net0: 192.168.0.84/255.255.255.0 gw 192.168.0.10
net0: fe80::5054:ff:fee9:a474/64
Next server: 192.168.0.10
Filename: d-i/n-a/bootnetx64.efi
tftp://192.168.0.10/d-i/n-a/bootnetx64.efi... ok
bootnetx64.efi : 229376 bytes [EFI]
```





# PXE boot: UEFI

## DHCP/TFTP Server:

```
$ journald -f
```

```
dnsmasq-dhcp: DHCPDISCOVER(enp2s0) 52:54:00:e9:a4:74
```

```
dnsmasq-dhcp: DHCPOFFER(enp2s0) 192.168.0.84 52:54:00:e9:a4:74
```

```
dnsmasq-dhcp: DHCPDISCOVER(enp2s0) 52:54:00:e9:a4:74
```

```
dnsmasq-dhcp: DHCPOFFER(enp2s0) 192.168.0.84 52:54:00:e9:a4:74
```

```
dnsmasq-dhcp: DHCPREQUEST(enp2s0) 192.168.0.84 52:54:00:e9:a4:74
```

```
dnsmasq-dhcp: DHCPACK(enp2s0) 192.168.0.84 52:54:00:e9:a4:74 debian034
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/bootnetx64.efi to 192.168.0.84
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/grub/x86_64-efi/normal.mod to 192.168.0.84
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/grub/x86_64-efi/extcmd.mod to 192.168.0.84
```

```
[...]
```

```
dnsmasq-tftp: sent /var/lib/tftpboot/d-i/n-a/grub/grub.cfg to 192.168.0.84
```

```
[...]
```



# Technisches Detail: Legacy BIOS – UEFI

## Woher weiß der Client welche Datei er laden soll?

Der DHCP server sendet den Pfad des *Network Bootstrap Program (NBP)* abhängig von der Identifizierung des Clients:

- ▶ Legacy BIOS
  - ▶ PXE Linux (pxelinux.0)
  - ▶ isc-dhcp: `option arch = 00:00`
  - ▶ dnsmasq: `dhcp-match=set:bios,option:client-arch,0`
- ▶ UEFI
  - ▶ GRUB (grubx64.efi, bootnetx64.efi)
  - ▶ isc-dhcpd: `option arch = 00:07`
  - ▶ dnsmasq: `dhcp-match=set:efi-x86_64,option:client-arch,7`



# Example 'installbox' (README.installbox)



## Preparations for the installbox:

```
## Install packages, stop dnsmasq and prevent it from starting at boot:
# apt install di-netboot-assistant dnsmasq nftables \
#                               debian-installer-netboot-amd64
# systemctl stop dnsmasq
# systemctl disable dnsmasq
# cp /usr/share/doc/di-netboot-assistant/examples/dnsmasq.conf.simple \
#                               /etc/dnsmasq.d/netboot-installer

## Prepare location for debian-installer-netboot-amd64 images:
# mkdir -p /var/lib/tftpboot/d-i/n-pkg/
```

## Example 'installbox' (README.installbox)

### Temporary convert to an installbox:

```
LAN_IF="${LAN_IF:-enp2s0}"
```

```
## Make packaged installer files available:
```

```
mount -o bind,ro /usr/lib/debian-installer/ /var/lib/tftpboot/d-i/n-pkg/  
di-netboot-assistant install daily
```

```
## Configure the netboot interface:
```

```
ip address add 192.168.0.10/24 dev $LAN_IF  
ip link set $LAN_IF up
```

```
## Enable forwarding:
```

```
sysctl -w net.ipv4.ip_forward=1
```

```
## Configure masquerading:
```

```
nft add table nat  
nft add chain nat prerouting { type nat hook prerouting priority 0 \; }  
nft add chain nat postrouting { type nat hook postrouting priority 100 \; }  
nft add rule nat postrouting masquerade
```

```
## Provide DNS, DHCP and a TFTP server for the netboot clients:
```

```
systemctl start dnsmasq
```

# Workshop with qemu built-in TFTP server

```
## PREPARE
sudo apt --no-install-recommends install di-netboot-assistant
sudo di-netboot-assistant install buster
TFTP_ROOT="/var/lib/tftpboot/"
qemu-system-x86_64 -enable-kvm -m 1024 -boot n -net nic \
    -net user,tftp=$TFTP_ROOT,bootfile=/d-i/n-a/pxelinux.0

## PRESEED
cd $TFTP_ROOT/d-i
sudo mkdir buster
sudo cp /usr/share/doc/di-netboot-assistant/examples/preseed.cfg buster/
sudo -e buster/preseed.cfg
sudo -e /etc/di-netboot-assistant/pxelinux.HEAD
sed -i "s/installbox/10.0.2.2/g" /etc/di-netboot-assistant/pxelinux.HEAD
sudo di-netboot-assistant rebuild-menu
qemu-system-x86_64 -enable-kvm -m 1024 -boot n -net nic \
    -net user,tftp=$TFTP_ROOT,bootfile=/d-i/n-a/pxelinux.0

## ANSIBLE
# a-pull preseeding: --url=https://salsa.debian.org/andi/debian-lan-ansible.git
qemu-img create -f qcow2 /tmp/vm.qcow2 8G
qemu-system-x86_64 -enable-kvm -m 1024 -boot n -net nic \
    -net user,tftp=$TFTP_ROOT,bootfile=/d-i/n-a/pxelinux.0,\
hostfwd=tcp::2222-:22,hostfwd=tcp::4443-:443 -hda /tmp/vm.qcow2
## READY!
```

