

Basis-Installation

Eine Schritt für Schritt Anleitung zur Installation eines openSUSE Leap wird es hier nicht geben. Daher hier nur ein paar Anmerkungen zum Festplatten-Management, der Software-Paket-Auswahl sowie der anschließenden Netzwerkkonfiguration.

Zur Installation eines invis-Servers ab Version 16.0 wird ein [openSUSE Leap 16.x](#) vorausgesetzt.

Einer der Unterschiede zu früheren Leap Versionen ist, dass sich ab 16.0 der Benutzer **root** per default nicht mehr per SSH am System anmelden kann. Daher wird empfohlen während des Setups einen lokalen Benutzer anzulegen der dann per **su** oder **sudo** arbeitet.

Software Auswahl

Der neue Installer Agama ist in seinen Möglichkeiten verglichen mit YaST extrem beschränkt. Schön daran ist aber, dass es im Grunde eine Webapplikation ist. D.h. Wenn der zu installierende Server in ihrem lokalen Netzwerk erreichbar ist, können Sie Agama im Browser Ihres Arbeits-PCs erreichen und bedienen.

Anders als in früheren Leap Versionen existiert die Rollenauswahl für das Setup im Agama-Installer nicht mehr.

...to be continued.

Festplatten-Management

Gerade im Hinblick auf das Datenträger-Management lässt **agama** einiges zu Wünschen übrig. Dumm für die Server-Installation, **agama** kann nicht mit Linux Software-RAID umgehen. Allerdings werden fertige RAID-Verbünde als Festplatten erkannt. Daraus folgt, dass ein SW-RAID gestütztes Setup mit **agama** nur funktioniert, wenn der oder die RAID-Verbünde bereits zuvor via Kommandozeile in einem Rettungssystem oder einem Terminal des unter **agama** laufenden Installationssystem erstellt wird.

Hinweis: Im Rettungssystem ist das Tastatur-Layout auf eine amerikanische Umgebung gesetzt. Dies lässt sich mit dem Kommando **loadkeys de** auf ein deutsches Layout ändern.

Dem Festplattenmanagement sollten Sie besondere Aufmerksamkeit widmen, schließlich geht es um sinnvolle Nutzung Ihres Plattenplatzes, der Sicherheit Ihrer Daten und der Wartbarkeit des Servers. Wir erläutern das Management beispielhaft anhand eines von uns in der Praxis meist genutzten Setups.

Wir gehen hier davon aus, dass Sie keinen Hardware-RAID-Controller im Einsatz haben, sondern statt dessen auf ein Linux-Software-RAID setzen. Vorteil dieser Methode ist auf jeden Fall, die Hardware-Unabhängigkeit sowie der Preisvorteil. Die Investition in einen Hardware-RAID-Controller macht sich hinsichtlich der höheren Performance bemerkbar.

Weiterhin gehen wir von einem einfachen Setup mit lediglich zwei Festplatten aus. Die Verwendung von mehr Festplatten und höheren RAID-Leveln läuft aber prinzipiell nach dem gleichen Schema ab.

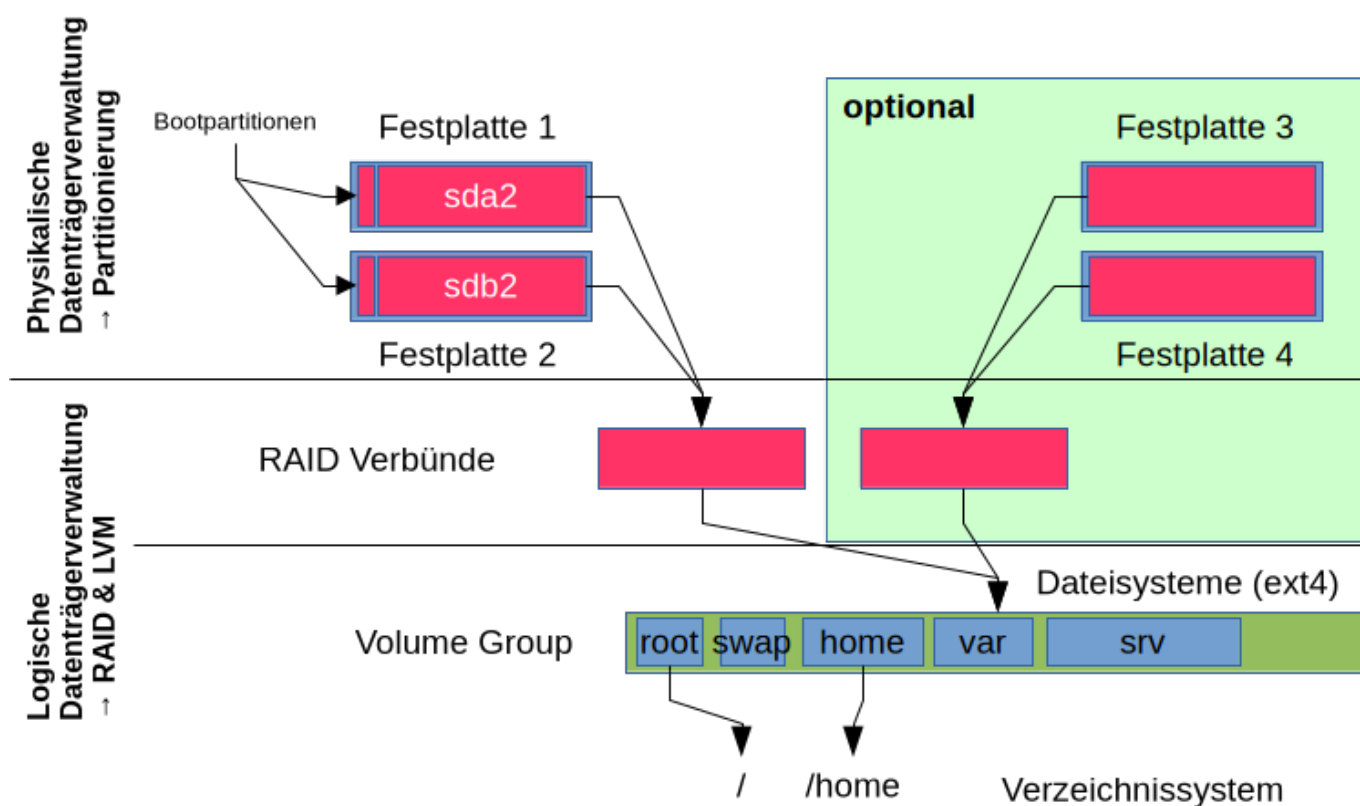
Unabhängig von der Größe der eingesetzten Festplatten bevorzugen wir eine GPT-basierte Partitionierung. Es hat sich gezeigt, dass dies im Falle eines Festplattendefekts weitaus weniger Probleme bereitet, als eine MBR basierte Partitionierung. Die Verwendung von GPT Partitionstabellen ist ab openSUSE Leap die Vorgabe, Sie müssen also nichts anpassen.

Ziel des Setups ist also eine GPT-basierte Partitionierung. Je nach BIOS/UEFI muss am Anfang jeder Platte entweder eine 8MB große Partition vom Typ „BIOS Boot“ (Legacy Boot) oder eine min. 100MB große EFI-Boot-Partition (UEFI-Boot) angelegt werden, in die Grub seine Boot-Records speichert. Der verbleibende Platz wird mit zwei Partitionen des Typs „Linux RAID“ belegt, die zu einem RAID1-Verbund kombiniert werden. Darauf aufbauend wird die Verteilung des zur Verfügung stehenden Platzes mittels Logical-Volume-Management (LVM) erledigt. Partitionen für Swap sind nicht erforderlich, dies wird in Form von LVM-Logical-Volumes umgesetzt. Wenn auch etwas gewöhnungsbedürftig, Agama beherrscht den Umgang mit LVM.

Hinweis: Wer sich statt dessen an einem vollständig manuellen Setup versuchen möchten findet [hier](#) eine nicht ganz aktuelle Anleitung.

Hinweis: Einige Funktionen des invis-Portals sowie des invis Server eigenen Backup-Tools gehen zwingend von der Kombination aus Software-RAID und LVM aus. Wählen Sie ein anderes Setup können diese Funktionen nicht oder nicht vollständig genutzt werden.

Einen Überblick über das für invis-Server angestrebte Datenträger-Layout, bietet folgende Grafik:



Hinweis: Lesen Sie für Systeme mit Festplatten größer 2TB und/oder aktiviertem UEFI-Boot bitte die entsprechenden Hinweise [hier](#).

Hier noch einmal die Befehle die Sie zur Partitionierung und zum Anlegen des RAID1 Verbundes benötigen:

Festplatte partitionieren:

```
linux:~ # fdisk /dev/sdX
```

Es werden wie gesagt auf jeder Platte 2 Partitionen benötigt, eine Boot-Partition (BIOS Boot oder UEFI Boot) und eine Partition vom Typ „LINUX RAID“.

RAID-Verbund anlegen:

```
linux:~ # mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda2 /dev/sdb2
```

Leider geht **agama** mit festen Größenangaben zu erstellender LVM-Volumes nicht korrekt um. Um sicher zu sein muss ein bereits angelegtes Volume noch ein mal editiert werden. Es zeigt sich, dass die Checkbox für „Größenzuwachs erlauben“ wieder aktiviert ist. Wird das übersehen nutzt **agama** den gesamten Platz des physical Volumes aus und vergrößert alle anzulegenden Volumes im Verhältnis der getätigten Größenangabe, was ziemlicher Schwachsinn ist.

Wird dieser Umstand nicht übersehen, ärgert **agama** mit dem nächsten Unfug. Er legt auf dem RAID-Volume eine Partition an, die genau so groß ist wie die Summe der logical Volumes. Den daraus resultierenden freien Platz nachträglich der angelegten Partition zuzuordnen ist so gut wie unmöglich. In diesem Fall ist der einfachste Workaround auf dem freien Platz mit **fdisk** eine zweite Partition anzulegen und diese dann als physical Volume der zuvor angelegten Volumegroup zuzuordnen.

Im Falle zu großer Volumes können diese, so sie mit Dateisystem **ext4** formatiert wurden, nachträglich verkleinert werden. Das funktioniert nur wenn die Dateisysteme **nicht** eingehängt sind, also im Falle eines var-Volumes nur aus einem Rescue-System heraus. Das ganze funktioniert wie folgt. Zunächst erfolgt der umount:

```
invis:~ # umount /dev/system/srv
```

Danach muss das Dateisystem zwingend überprüft werden:

```
invis:~# e2fsck -f /dev/system/srv
```

Jetzt kann zunächst das Dateisystem verkleinert werden, im Beispiel auf 100GiB:

```
invis:~# resize2fs /dev/system/srv 100G
```

Anschließend wird das Volume passend dazu verkleinert:

```
invis:~# lvresize -L 100G /dev/system/srv
```

Danach kann das Volume wieder eingehängt werden. Am einfachsten mit:

```
invis:~# mount -a
```

Achtung: Das Dateisystem **XFS** lässt sich nicht verkleinern.

Wird das Verkleinern in einem Rescue-System vorgenommen sind zunächst MD-RAID-Verbünde und auch darauf befindliche Logical-Volumes inaktiv. Beides muss erst aktiviert werden. Zunächst der

RAID-Verbund mit:

```
invis:~ # mdadm --assemble --scan
```

Dann sind die betreffenden Volumes zu aktivieren, hier am Beispiel des var-Volumes:

```
invis:~ # lvchange -ay /dev/system/var
```

Danach wird verfahren wie zuvor gezeigt.

Hoffen wir mal, dass **agama** irgendwann erwachsen wird!

Letzte Vorbereitungen

Um das invis-Setup einzuleiten benötigen Sie unser Setup-Paket „invisAD-setup“. Dieses Paket ist nicht in den Standard-Repositories enthalten. Es muss also ergänzend eines unserer Repositories eingebunden werden.

Zur Verfügung stehen folgende Repositories zur Verfügung:

1. **spins:invis:stable** - Stabile Version der invis-Server Setup Pakets. Nutzen Sie dieses Repository für produktiv genutzte invis-Server
2. **spins:invis:unstable** - In Entwicklung befindliche Versionen der invis-Server Setup Pakets. Nutzen Sie dieses Repository, wenn Sie uns mit Rat, Tat, Lob oder Kritik bei der Weiterentwicklung unterstützen möchten.

Zur Einbindung des gewünschten Repositories haben wir mit **invisprep** ein Script erstellt, welches diesen Schritt automatisch durchführt.

Download: [invisprep](#)

Laden Sie es auf Ihren Server herunter, entpacken Sie es und führen Sie es aus.

Hinweis: Beim direkten Download der Datei mit **wget** ändert sich der Name der Datei. Das kann beim Entpacken zu Verwirrung führen. Dabei hilft folgende Kommandozeile:

```
linux:~ # wget -O invisprep.gz  
https://wiki.invis-server.org/lib/exe/fetch.php?media=invis_server_wiki:installation:invisprep.gz
```

Die Datei kann jetzt entpackt und ausgeführt werden:

```
invis:~ # gunzip invisprep.gz  
invis:~ # chmod +x invisprep  
invis:~ # ./invisprep
```

Danach kann das invis-Setup Paket installiert werden:

```
linux:~ # zypper ref
linux:~ # zypper in invisAD-setup-16
```

Seit Version 11 des invis-Servers ist die Major-Release-Nummer teil des Paketnamens. Sie müssen sie natürlich korrekt angeben. Es ist beispielsweise möglich, dass speziell im „unstable“ Repository mehrere Versionen vorhanden sind.

Hinweis: Dass bei der Installation des invisAD-setup RPMs sehr viele weitere Software-Pakete installiert werden ist normal. 😊

Netzwerkconfiguration

Durch den Wegfall von YaST mit Leap 16 hat sich hier einiges geändert. Auch „wicked“ als Dienst zum Management der Netzwerkschnittstellen ist nicht mehr Teil von openSUSE Leap. openSUSE Leap setzt zunächst voll und ganz auf den Networkmanager. Dieser spielt seine Stärken vor allem auf Desktop-Installationen aus, da es damit normalen Usern ohne administrative Rechte gestattet ist beispielsweise einen WLAN-Zugriff zu konfigurieren. Für Server-Installationen ohne grafische Oberfläche ist er unserer Meinung nach eher nicht geeignet. Daher ersetzen wir ab invis-Server 16, den Networkmanager, respektive „wicked“ durch „systemd-networkd“, eine Komponente des Systemd, wie unschwer am Namen zu erkennen ist.

Der Wechsel erfolgt automatisch mit dem Ausführen des Scripts **netsetup**.

Unverändert bleibt, dass beim invis-Server die einzelnen Netzwerkschnittstellen entsprechend Ihrer Verbindung, bzw. Firewall-Zone benannt werden.

Um die Benennung der Netzwerkschnittstellen mit den einzelnen Firewall-Zonen eines invis-Servers in Verbindung zu bringen, werden die Namen der Netzwerkschnittstellen auf Basis von udev-Regeln vergeben. Auch die Benennung der im Laufe des Setups einzurichtenden VPN-Schnittstelle wird ein entsprechender Name gegeben:

Vom System vergebener Geräte name	Aktueller Name
eth0 oder enpXsY	extern
eth1 oder enpXsY	intern
eth2 oder enpXsY	dmz
tun1	vpn

Die erste Netzwerkkarte des Systems (extern) stellt die Verbindung des Servers mit dem Internet her - entspricht somit der externen Zone Ihrer Firewall. Je nach dem, wie der dem invis-Server vorgeschaltete Router konfiguriert ist, müssen Sie „extern“ als DHCP-Client oder gemäß den Vorgaben Ihres Providers konfigurieren.

Die zweite Netzwerkkarte (intern) muss mit einer festen IP-Adresse versehen werden. Selbst, wenn über den Internet-Service-Provider eine feste IP-Adresse zur Verfügung steht, ist für das interne Netz die Verwendung eines eigenen Netzes zwingend.

Falls mehr als zwei Netzwerkschnittstellen vorhanden sind, wird die dritte Schnittstelle in **dmz** umbenannt, auch wenn das invis-Server Setup dies (derzeit) in keiner Weise nutzt.

Die Erstellung der UDEV-Regeln wird mittels unseres Script **netsetup** erledigt, aufgerufen wird es

ohne weitere Parameter:

```
localhost:~ # netsetup  
localhost:~ # reboot
```

Im Anschluss an dessen Ausführung muss das System neu gestartet werden. **netsetup** sorgt automatisch dafür, dass die externe Schnittstelle versucht sich eine IP-Adresse per DHCP zu besorgen. Wenn alles gut geht, ist der Server also nach dem Reboot per **SSH** erreichbar.

Die Konfiguration der internen oder weiterer Schnittstellen erfolgt mittels des neuen Scripts **netdevconf**.

Achtung: Vermeiden Sie es Ihrem lokalen Netzwerk typisch Adressbereiche gängiger Router-Modelle wie auch üblicher Docker Konfigurationen zu verpassen. Hier ein paar Beispiele von denen Sie die Finger lassen sollten:

typische IP Netze gängiger Router und Docker-Installationen
192.168.0.0/24
192.168.1.0/24
192.168.2.0/24
192.168.50.0/24
192.168.100.0/24
192.168.177.0/24
192.168.178.0/24
192.168.188.0/24
172.16 - 24.0.0/16

Achtung: Bevor Sie jetzt die Netzwerkschnittstellen Ihres invis-Servers konfigurieren ein Hinweis dazu. invis-Server können lediglich mit 16 und 24 Bit breiten Netzwerkmasken also „255.255.0.0“ (/16) und „255.255.255.0“ (/24) umgehen.

Idealerweise konfigurieren Sie für die interne Netzwerkschnittstelle ein privates IP-Netzwerk der Klassen „B“ (172.16.0.0 bis 172.31.255.255) oder „C“ (192.168.0.0 bis 192.168.255.255). Über die Unterstützung von Klasse „A“ Netzen denken wir noch nach, erachten dies aber nicht wirklich als notwendig für „kleine“ Netze.

netdevconf erfordert als Aufrufparameter zunächst den Namen der Schnittstelle und dann die gewünschte IP-Adresse nebst Netzwerkmaske in Kurzschreibweise:

```
localhost:~ # netdevconf intern 172.26.0.10/16
```

Wir unterteilen die Netze der beiden unterstützten Netzwerkklassen für den DHCP-Server in verschiedene Bereiche (Damit ist **kein** Subnetting gemeint). Die nachfolgende Tabelle zeigt die verschiedenen Bereiche, angezeigt wird jeweils nur der Host-Anteil der IP-Adressen.

Gerätekategorie	Klasse C Netz	Klasse B Netz
Server	.11 - .19	.0.11 - .0.253
Drucker	.20 - .50	.1.1 - .1.254
IP-Geräte	.60 - .90	.2.1 - .3.254

Geräteklasse	Klasse C Netz	Klasse B Netz
PCs	.120 - .199	.4.1 - .4.254
dyn. Bereich	.200 - .220	.200.1 - .200.254

Anders als früher mit YaST wird der Hostname zu diesem Zeitpunkt noch nicht konfiguriert. Dies ist in das nachfolgende Setup mit **sine2** gewandert.

Achtung: Ab openSUSE Leap 16.0 hat Leap das Security-Framework gewechselt. Es kommt jetzt SELinux anstelle von AppArmor zum Einsatz. Wir arbeiten daran unser invis-Server Setup kompatibel mit SELinux zu gestalten. Das ist allerdings nicht ganz so trivial. Wir empfehlen daher vorübergehend SELinux von **enforcing** auf **permissive** umzustellen.

Die Einstellung wird in

```
/etc/selinux/config
```

vorgenommen:

```
...  
SELINUX=permissive  
...
```

Damit ist die Vorarbeit abgeschlossen, weiter geht'S mit **sine2** und dem eigentlichen Server-Setup.

From:
<https://wiki.invis-server.org/> - **invis-server.org**

Permanent link:
https://wiki.invis-server.org/doku.php?id=invis_server_wiki:installation:basesetup-160&rev=1778497825

Last update: **2026/05/11 11:10**

