invis-Server Upgrade von 13.5 auf 14.1

Richtig gelesen, wir lassen invis-Server Version 14.0 aus. Begründet liegt dies in den unverhofft aufgetretenen Probleme mit den in openSUSE Leap enthaltenen Samba-Paketen. Zum Hintergrund. Mit openSUSE Leap 15.0 wurden seitens openSUSE Samba-Pakete mit ActiveDirectory-Domain-Controller Funktion ausgeliefert. Darauf haben wir lange gewartet. Diese nutzen jedoch statt der von den Samba-Entwicklern integrierten Kerberos-Implementation Heimdal, den aus der Distribution stammenden MIT-Kerberos-Server.

Wie sich in der Praxis herausstellte, war das ein riesiges Problem. Kerberos funktioniert damit einfach nicht korrekt. Kurz wir haben in den sauren Apfel gebissen und begonnen wieder eigene Samba-Pakete mit Heimdal-Kerberos zu bauen. Lesen Sie dazu auch unseren Blog-Beitrag **Rolle Rückwärts** um die lange Geschichte zu erfahren.

Weiterhin basiert invis-Server 14.1 bereits auf openSUSE Leap 15.1, d.h. es müssen 2 Distributionsupgrades in Folge gemacht werden.

Hinweis: *Die folgende Anleitung setzt voraus, dass Ihr Server* **definitiv** *auf dem Stand von Version 13.5 ist, also bereits PHP7 und ownCloud 10.x verwendet. Ist das nicht der Fall müssen Sie zunächst auf Version 13.5 aktualisieren. Eine Anleitung dazu finden Sie* ***hier****.*

Vorbereitung

Sichern Sie alle Datenbanken des Servers. Sie können dafür die Tools des invis-Servers nutzen:

ActiveDirectory

```
invis:~ # adbackup
```

Kopano

Führen Sie hier beide Sicherungswege durch:

```
invis:~ # kdbdump
...
invis:~ # kbackup
```

Weitere Datenbanken

```
invis:~ # alldump
```

Dokuwiki

```
invis:~ # dwdatasnapshot
```

Da mit der neuen auf MIT-Kerberos basierenden Samba-Version 4.7. die im Laufe des Upgrades installiert wird, werden überarbeitete AppArmor Profile notwendig. Um negative Effekte auf das

Upgrade durch mögliche AppArmor-Blockaden vorzubereiten wird AppArmor vor dem Upgrade deaktiviert.

```
invis:~ # systemctl stop apparmor.service
...
invis:~ # systemctl disable apparmor.service
```

Im späteren Verlauf des Upgrades wird AppArmor mit neuen Profilen wieder aktiviert.

Auch der Email-Abruf sollte vor dem Upgrade deaktiviert werden:

```
invis:~ # systemctl stop fetchmail.service
invis:~ # systemctl disable fetchmail.service
```

Auch der Samba Domain-Controller wird abgeschaltet und deaktiviert.

```
invis:~ # systemctl stop samba.service
invis:~ # systemctl disable samba.service
```

Die neue Samba-Version bringt ohnehin ein eigenes Service-Unit-File mit.

Samba Upgrade

Beginnen wir mit dem Herz des Servers, dem ActiveDirectory.

Achtung: *Stellen Sie sicher, dass alle Client-PCs bevor Sie beginnen herunter gefahren sind!*

Entfernen Sie zunächst das bisherige Samba-Repository. Ermitteln Sie dazu die Nummer des Repositories:

```
invis:~ # zypper repos | grep samba
22 | spins_invis_stable_samba46      | Samba AD Packages
(openSUSE_Leap_42.3)                          | Ja        | (r ) Ja
| Ja
invis:~ #
```

im Beispiel trägt das Repo die Nr. 22. Mit der Nummer kann es jetzt entfernt werden:

```
invis:~ # zypper rr 22
```

Speziell für das Upgrade haben wir eigens ein Samba-Repository erzeugt, welches aktuelle Samba-Pakete auch für openSUSE Leap 42.3 bereit stellt. Die darin enthaltenen Pakete sind nicht für den Dauerbetrieb gedacht, da es kompatibilitätsprobleme mit dem SSSD-Dienst gibt. Die Pakete darin ermöglichen lediglich das Upgrade von Samba auf die Version, die auch von invis-Server 14.1 genutzt wird.

Jetzt können Sie das neue „temporäre“ Samba-Repository hinzufügen und den Repository-Cache

auffrischen:

```
invis:~ # zypper ar
https://download.opensuse.org/repositories/spins:/invis:/13.5:/samba/openSUS
E_Leap_42.3/spins:invis:13.5:samba.repo
invis:~ # zypper ref
```

Jetzt können die neuen Samba-Pakete installiert werden. Es wird dabei direkt von Samba-Version 4.6.x auf 4.10.x aktualisiert. Dies erfordert ein wenig Nacharbeit.

Ermitteln Sie zunächst die Nummer des neuen Samba-Repositories:

```
invis:~ # zypper repos | grep samba
20 | spins_invis_13.5_samba     | Samba 4.10 with Heimdal Kerberos
(openSUSE_Leap_42.3)             | Ja        | (r ) Ja         | Nein
invis:~ #
```

Im Beispiel trägt das neue Repository die Nr. 20, daraus ergibt sich das Kommando zum Paket-Upgrade:

```
invis:~ # zypper dup --from 20 --allow-vendor-change
```

Dieses Kommando löst eine Reihe Paketkonflikten aus.

Wählen Sie jeweils Lösung **2**:

```
...
 Lösung 2: Deinstallation von libdcerpc-server0-4.6.16-103.19.x86_64
...
```

```
...
  Lösung 2: Deinstallation von samba-ad-4.6.16-103.19.x86_64
...
```

```
...
 Lösung 2: Deinstallation von libsamba-policy0-4.6.16-103.19.x86_64
...
```

```
...
 Lösung 2: Deinstallation von libsmbldap0-4.6.16-103.19.x86_64
...
```

Installieren Sie jetzt noch fehlende Samba-Pakete nach:

```
invis:~ # zypper in samba-ad-dc ldb-tools
```

Sind alle Pakete installiert, darf Samba noch **nicht** wieder gestartet werden. Mit Samba 4.10. hat sich die Verzeichnisstruktur unter

```
/var/lib/samba
```

moderat geändert. D.h. die zuvor angelegte Datensicherung muss in die neue Verzeichnisstruktur wiederhergestellt werden. Für diesen Zweck haben wir ein Script entwickelt. Zwar ist dieses Script im aktuellen invisAD Setup Paket enthalten, leider stehen dieses Paket erst nach Umstrukturierung der Software-Repositories wie nachfolgend beschrieben zur Verfügung. Um das Samba-Upgrade jetzt dennoch abschließen zu können stellen wir das Script auch hier zum direkten Download zur Verfügung. Laden Sie die Datei wie folgt auf Ihren Server herunter:

```
invis:~ # wget -O upgradead.gz
http://wiki.invis-server.org/lib/exe/fetch.php/invis_server_wiki:upgradead.g
z
```

Entzippen Sie die Datei:

```
invis:~ # gunzip upgradead.gz
```

...und machen Sie sie ausführbar:

```
invis:~ # chmod +x upgradead
```

Jetzt können Sie damit Ihr AD wiederherstellen. Sie benötigen dazu den Pfad zur AD-Sicherung. Sie finden die Sicherungen und

```
/srv/shares/archiv/sicherungen/vollsicherungen/ad/
```

Sie benötigen von dort die aktuellste Datei, was anhand des Datums im Dateinamen leicht zu erkennen ist. Führen Sie das Script wie folgt aus:

```
invis:~ # ./upgradead
/srv/shares/archiv/sicherungen/vollsicherungen/ad/Samba_20190815-075033.tar.
gz
```

Zur Fehlervermeidung stellt das Script nach der Ausführung eine „Sind Sie sicher?“ Abfrage, die zu bejahen ist. Das Script startet den Samba-AD-DC Dienst automatisch wieder. Sie können jetzt verschiedene Tests durchführen. Z.B.: Anmelden an einem Windows-PC, Anmelden am invis-Portal, DNS-Abfragen mit ***dig*** usw.

Klappt alles ist das Samba-Upgrade muss der neue Samba-Domain-Controller Dienst zum automatischen Start vorgesehen werden.

```
invis:~ # systemctl enable samba-ad-dc.service
```

Damit ist das Samba-Upgrade abgeschlossen.

Distributions-Upgrade Sprung 1

Hinweis: *In einzelnen Fällen gab es bei unseren Tests beim Sprung von openSUSE Leap 42.3 auf 15.0, dann ein Problem wenn das System als virtuelle Maschine betrieben wurde. Der aktuelle Kernel*

von Leap 15.0 scheint mit Virtualisierung ein Problem zu haben. Die Folge war, dass das System ewig zum starten brauchte und danach nicht benutzbar war. Umgehen lässt sich das Problem in dem das System über den noch vorhandenen Kernel von Leap 42.3 gestartet wird und dann das nächste Distributionsupgrade auf 15.1 durchgeführt wird. Mit dem Kernel dieser Version treten die Probleme nicht mehr auf.

Mit dem Sprung auf invis-Server 14.0 muss auf openSUSE Leap 15.0 aktualisiert werden. Die Vorgehensweise dazu ist denkbar einfach. Zunächst müssen Sie Ihre Software-Repositories daran anpassen. Es kann auch nicht schaden zunächst mal alles zu sichern, so wie es ist:

```
invis:~ # cp -R /etc/zypp/repos.d /etc/zypp/repos.d.bak
```

Prüfen wir jetzt, ob ein CD/DVD Repository bei der Installation verwendet wurde. Ist das der Fall, kann es gelöscht werden:

```
invis:~ # grep "cd://" /etc/zypp/repos.d/*
/etc/zypp/repos.d/openSUSE-42.3-0.repo:baseurl=cd:///?devices=/dev/disk/by-
id/ata-TSSTcorp_CDDVDW_SH-224BB_R8WS68BCB00TYX
invis:~ # rm /etc/zypp/repos.d/openSUSE-42.3-0.repo
```

Weiterhin verfügt Ihr invis-Server über zwei weitere Repositories unseres Projektes. Da wir für Version 14.x des invis-Servers eine neue Repository-Struktur aufgebaut haben, können diese beiden Repositories entfernt und dann durch die neuen Repositories ersetzt werden:

```
invis:~ # zypper repos |grep spins_invis
20 | spins_invis_13.5_samba     | Samba 4.10 with Heimdal Kerberos
(openSUSE_Leap_42.3)             | Ja        | (r ) Ja         | Nein
21 | spins_invis_common             | Common packages for invis-Server
stable & unstable (openSUSE_42.3) | Ja        | (r ) Ja         | Ja
22 | spins_invis_stable             | Stable Packages for invis-servers
(openSUSE_Leap_42.3)             | Ja        | (r ) Ja         | Ja
```

Zu entfernen sind also die Repositories Nr. 21 und 22:

```
invis:~ # zypper rr 22
Repository 'Production Project for the openSUSE invis-Server Spin
(openSUSE_Leap_42.3)' entfernen
.....................................[fertig]
Repository 'Production Project for the openSUSE invis-Server Spin
(openSUSE_Leap_42.3)' wurde entfernt.
invis:~ # zypper rr 21
Repository 'Common packages for invis-Server stable & unstable
(openSUSE_42.3)' entfernen
.........................................[fertig]
Repository 'Common packages for invis-Server stable & unstable
(openSUSE_42.3)' wurde entfernt.
invis:~ # zypper rr 20
Repository 'Samba 4.10 with Heimdal Kerberos  (openSUSE_Leap_42.3)'
entfernen .....................................[fertig]
Repository 'Samba 4.10 with Heimdal Kerberos  (openSUSE_Leap_42.3)' wurde
entfernt.
```

```
invis:~ #
```

Fügen wir jetzt die neuen Repositories hinzu:

```
invis:~ # zypper ar
https://download.opensuse.org/repositories/spins:/invis:/15:/common/openSUSE
_Leap_15.0/spins:invis:15:common.repo
...
invis:~ # zypper ar
https://download.opensuse.org/repositories/spins:/invis:/15:/stable/openSUSE
_Leap_15.0/spins:invis:15:stable.repo
...
invis:~ # zypper ar
https://download.opensuse.org/repositories/spins:/invis:/15:/stable:/samba/o
penSUSE_Leap_15.0/spins:invis:15:stable:samba.repo
```

Wenn Sie Kopano auf Basis einer offiziellen Kopano-Subskription einsetzen müssen Sie eine der Repository-Dateien manuell anpassen. Editieren Sie die Datei:

```
/etc/zypp/repos.d/Kopano-openSUSE_limited.repo
```

, indem Sie die mit „baseurl=“ beginnende Zeile auf folgende URL abändern:

```
[Kopano-openSUSE-15.0_limited]
name=Kopano-openSUSE-15.0_limited
enabled=1
autorefresh=1
baseurl=https://download.kopano.io/limited/core:/final/openSUSE_Leap_15.0
path=/
type=rpm-md
keeppackages=0
```

Dieses Kopano-Repository verlangt Zugangsdaten, diese wurden während des Server-Setups eingegeben und in

```
/root/.zypp/credentials.cat
```

hinterlegt. Sie können in dieser Datei ebenfalls die URL zum Repository anpassen oder die Zugangsdaten beim nächsten ***zypper ref*** Kommando erneut eingeben. In der genannten Datei muss die erste Zeile geändert werden:

```
[https://download.kopano.io/limited/core:/final/openSUSE_Leap_15.0/]]
username = FSP Computer und Netzwerke
password = supergeheim
...
```

Abschließend sind noch die openSUSE Versionsnummern in den verbleibenden Repository-Dateien zu ersetzen:

```
invis:~ # sed -i 's/42\.3/15\.0/g' /etc/zypp/repos.d/*
```

Jetzt kann das Distributions-Upgrade durchgeführt werden:

```
invis:~ # zypper ref
...
invis:~ # zypper dup
```

Stimmen Sie dem Vorschlag den ***zypper dup*** macht zu, danach beginnt das Paket-Upgrade.

Hinweis: *Je nach Stand Ihrer Kopano-Installation, kann es sein, dass das Upgrade einige Paketkonflikte mit sich bringt. Lösen Sie diese Pakete in dem Sie jeweils die Deinstallation der älteren Pakete (vermutlich Version 8.6.9) als Lösung auswählen. In unseren Tests war dies in der Regel Lösung 2.*

Hinweis: *Weiterhin wird **zypper** einige Dateikonflikte melden. Erlauben Sie das Ersetzen der Dateien mit „ja“ um das Distributionsupgrade abzuschließen.*

Mit dem Distributionsupgrade wird eine aktuelle Version von MariaDB installiert, was ein Upgrade der Tabellenstrukturen erfordert. Normalerweise geschieht dies mit einem Neustart des Dienstes automatisch. Je nach Datenbankgröße kann dies aber so lange dauern, dass systemd die Geduld verliert und in einen Timeout läuft. Daher sollte das Upgrade manuell durchgeführt werden:

```
invis:~ # /usr/lib/mysql/mysql-systemd-helper upgrade
```

Starten Sie den Server jetzt neu.

Distributions-Upgrade Sprung 2

Der Sprung von openSUSE Leap 15.0 nach 15.1 ist deutlich unkomplizierter als der vorherige Schritt. Passen Sie zunächst die Repositories an:

```
invis:~ # sed -i 's/15\.0/15\.1/g' /etc/zypp/repos.d/*
```

Jetzt kann das Upgrade gestartet werden:

```
invis:~ # zypper ref
...
invis:~ # zypper dup
```

Diesmal treten keine Paketkonflikte auf, ggf. aber wieder Dateikonflikte. Erlauben Sie das überschreiben der alten Dateien mit „ja“. Danach starten Sie das System neu.

invis-Setup aktualisieren

Durch den Wechsel der Repository-Struktur für invis-Server ab Version 14.0 wird beim Distributionsupgrade das invis-Server Setup-Paket gelöscht. Dies ist jetzt neu zu installieren:

```
invis:~ # zypper in invisAD-setup-14
```

Nach der Installation sind zunächst ein paar Anpassungen der Konfiguration erforderlich, zunächst sind allerdings die vorherigen Konfigurationen wieder herzustellen:

```
invis:~ # old /etc/invis/invis.conf
invis:~ # cp /etc/invis/invis.conf.rpmsave /etc/invis/invis.conf
...
invis:~ # old /etc/invis/invis-pws.conf
invis:~ # cp /etc/invis/invis-pws.conf.rpmsave /etc/invis/invis-pws.conf
```

Für die neue Version des Samba-Servers haben wir neue Apparmor-Profile erstellt, die Sie installieren müssen:

```
invis:~ # cp /usr/share/sine/templates/samba_ad/apparmor/* /etc/apparmor.d/
```

Starten Sie Apparmor jetzt neu:

```
invis:~ # systemctl restart apparmor.service
```

Konfigurationsanpassung invis Server

Passen Sie jetzt in der Haupkonfigurationsdatei

```
/etc/invis/invis.conf
```

die Versionsnummer des invis-Server-Setups an:

```
#invis-server Version
invisVersion:14.1
```

Suchen Sie jetzt in der genannten Datei nach einer mit der Direktive „avCheck“ beginnenden Zeile und ändern Sie sie wie folgt ab:

```
# [all|profiles|none]
avCheck:none
```

Statt nur „ja“ oder „nein“ bezogen auf regelmäßige Virenprüfungen im Fileserver, können Sie jetzt entscheiden, ob alles, nichts oder lediglich die Benutzerprofile geprüft werden.

Gehen Sie an eine regelmäßigen Virenprüfung vorsichtig heran. Je nach Datenmenge laufen diese Prüfungen sehr lange und erzeugen hohe Systemauslastungen. Wir nutzten dieses Feature in der Praxis oft gar nicht, viel wichtiger sind aktuelle und aktive Virenscanner auf den Clients.

Jetzt ist die Datei noch mit neuen Konfigurationen zu ergänzen. Fügen Sie folgende Blöcke hinzu:

```
# Disk Warranty Time - Garantiezeitraum der eingesetzten Festplatten
```

```
# 5 Jahre = 43800 Stunden (Gilt für die meisten 24/7 Festplatten)
# 3 Jahre = 26280 Stunden (Gilt für gute Consumer Festplatten)
# 1 Jahr = 14140 Stunde (Gilt für Low-Budged Festplatten)
diskWarrantyTime:43800
```

Das regelmäßig laufende Tool ***diskchecker*** überprüft jetzt auch die absolvierten Betriebsstunden der eingesetzten Festplatten und setzt diese mit der vom Hersteller garantierten Laufzeit in Beziehung. Passen Sie die Konfiguration entsprechend den von Ihnen eingesetzten Festplatten an.

```
# Pfad zu den Verzeichnisvorlagen der Gruppenverzeichnisse
groupDirTemplatePath:/srv/shares/media/portal/verzeichnisvorlagen
```

Hier müssen Sie nichts weiter unternehmen. Beim Anlegen von Gruppen mit Gruppenarbeitsverzeichnissen, können diese jetzt auf Basis von Verzeichnisvorlagen erzeugt werden.

Konfigurationsanpassung invis Portal

Die Anpassungen der invis-Portal Konfiguration sind zwar nicht zwingend, sollten aber dennoch durchgeführt werden. Anzupassen ist die Datei

```
/etc/invis/portal/config.php
```

Passen Sie zunächst die Versionsnummern am Anfang der Datei an:

```
...
$GROUPWARE = 'kopano';
$INVISVERSION = '14.1';
$OPENSUSEVERSION = '15.1';
...
```

Jetzt sind auch hier ein paar neue Konfigurationen hinzuzufügen. Nachfolgend sind die Zeilen immer mit der bereits vorhandenen vorhergehenden Zeile gezeigt:

```
...
 $SFU_GUID_BASE = '20000';
 $GROUP_DIR_TEMPLATE_PATH = '/srv/shares/media/portal/verzeichnisvorlagen';
...
```

In der bereits vorhandenen Konfigurationsdirektive „$SMB_GROUPSTOEXTEND“ ist der Wert „diradmins“ hinzuzufügen:

```
...
$SMB_GROUPSTOEXTEND = array("Domain Users", "Domain Admins", "Domain
Guests", "Archiv", "Verwaltung", "diradmins");
...
```

Am Ende des SMB-Blocks ist folgende Zeile zu ergänzen:

```
...
```

```
 $SMB_FILESERVER = strtoupper("null");
 $SMB_DEFAULT_LOGON_SCRIPT = ("user.cmd");
...
```

In der Liste der via invis-Portal zu verwaltenden Dienste ist der neue Firewall-Daemon einzufügen:

```
...
        array('fetchmail','Emails abholen'),
        array('firewalld','Firewall'),
        array('freshclam', 'Virenscanner Updater')
...
```

Weiterhin hat sich der Name des Domaincontroller-Daemons geändert. Aus:

```
...
        array('samba', 'Active Directory'),
...
```

wird:

```
...
        array('samba-ad-dc', 'Active Directory'),
...
```

Damit ist auch die Konfiguration des invis-Portals auf Stand.

Legen Sie abschließend im Administrationsbereich des invis-Portals die Gruppe „diradmins“ an, sie benötigt **kein** Gruppenarbeitsverzeichnis und sollte vom Typ „Team“ sein.

Mitglieder der Gruppe haben die Berechtigung Verzeichnisvorlagen für die Gruppenarbeitsverzeichnisse anzulegen und zu bearbeiten.

Zugriffsrechte der Netzwerkfreigaben wiederherstellen

Dieser Schritt wird ainfach mit dem Toolbox-Script ***afterup*** erledigt:

```
invis:~ # afterup
```

Damit ist das invis-Server Setup auf aktuellem Stand.

Inbetriebnahme des Firewall-Daemons (firewalld)

Mit Erscheinen der openSUSE Leap Version 15.0 ist SUSE vom jahrelang selbst entwickelten Firewall-System „**SuSEfirewall2**“ auf den von RedHat entwickelten „**firewalld**“ umgestiegen. Diesen Umstieg müssen Sie mit Hilfe von ***sine2*** vollziehen.

Bevor es los geht, sollten Sie dafür sorgen, dass die SuSEfirewall2 nach einem Reboot nicht mehr gestartet wird:

```
invis:~ # systemctl disable SuSEfirewall2.service
```

...und deaktivieren Sie sie:

```
invis:~ # systemctl disable SuSEfirewall2.service
```

Um das sine2-Firewall-Modul aufrufen zu können bedarf es eines Tricks. Normalerweise ist dieses Modul kein optionales Modul, kann also nicht einzeln aufgerufen werden. Genau das muss für den nächsten Schritt geändert werden. Öffnen Sie dazu in Ihrem Editor folgende Datei:

```
/usr/share/sine/registered-modules.txt
```

Ändern Sie darin die Zeile:

```
...
13:d:firewall
...
```

auf

```
...
13:o:firewall
...
```

ab.

Jetzt können Sie das Firewall-Modul direkt aufrufen. Es kümmert sich um alles weitere:

```
invis:~ # sine2 firewall
```

Damit ist auch der neue Firewall-Daemon in Betrieb genommen.

Kopano wieder in Betrieb nehmen

Stoppen Sie zunächst ggf. laufende Kopano-Dienste:

```
invis:~ # runkopano stop
```

Nach den verschiedenen Upgrades fehlt Kopano unter Umständen das Recht auf seine Schlüsseldatei

```
/etc/invis/certs/kopano.pem
```

zuzugreifen. Evtl. befindet sich in der Konfiguration des Serverdienstes noch ein falscher Pfad, der Systembenutzer „kopano“ muss mittlerweile Mitglied der Gruppe „pkeys“ sein und die Datei „kopano.pem“ befindet sich noch gar nicht am genannten Ort. Abhängig ist dies vom Ausgangspunkt

des Upgrades.

Letzteres lässt sich einfach beheben:

```
invis:~ # mkkopanokey
```

Diesen Befehl können Sie einfach auf Verdacht ausführen, er richtet keinen Schaden an.

Um Kopano in die Gruppe „pkeys“ aufzunehmen bearbeiten Sie die Datei

```
/etc/groups
```

unter Verwendung des Kommandos ***vigr*** (vi Kenntnisse vorausgesetzt!). Suchen Sie die Zeile „pkeys“ und ergänzen Sie sie wie folgt:

```
...
pkeys:x:998:wwwrun,kopano
...
```

Kontrollieren Sie jetzt noch in

```
/etc/kopano/server.cfg
```

in Zeile (ca.) 195 ob der korrekte Pfad (siehe oben) angegeben ist.

Testen Sie jetzt, ob sich der Kopano-Serverdienst starten lässt:

```
invis:~ # systemctl start kopano-server.service
```

Kontrollieren Sie dies einfach mit:

```
invis:~ # systemctl status kopano-server.service
```

Läuft der Server-Dienst, müssen Anpassungen an der Datenbank vorgenommen werden. Dies erledigt das Kommando:

```
invis:~ # kopano-dbadm usmp
```

Je nach Datenbankgröße kann das eine ganze Weile dauern. Starten Sie im Anschluss daran den Kopano-Server-Dienst neu:

```
invis:~ # systemctl restart kopano-server.service
```

Kontrollieren Sie das Ende der Kopano-Server Logdatei auf Fehler:

```
invis:~ # less /var/log/kopano/server.log
```

Im Idealfall ist alles OK.

Nach dem Upgrade fehlt für den Kopano-Search Dienst ein Python3 Paket, installieren Sie es wie folgt nach:

```
invis:~ # zypper in python3-magic
```

Danach sollte sich auch dieser Dienst starten lassen:

```
invis:~ # systemctl start kopano-search.service
```

Auch dies sollten Sie überprüfen:

```
invis:~ # systemctl status kopano-search.service
```

Der Kopano-Gateway Dienst benötigt eine neue Konfigurationsdatei. Hier können Sie einfach unsere Vorlage aus dem invis-Setup-Paket nutzen. Sichern Sie zuvor vorsichtshalber die bestehende Datei:

```
invis:~ # old /etc/kopano/gateway.cfg
moving /etc/kopano/gateway.cfg to /etc/kopano/gateway.cfg-20190829
invis:~ # cp /usr/share/sine/templates/groupware/kopano/gateway.cfg
/etc/kopano/
```

Starten Sie jetzt den Gateway-Dienst und kontrollieren Sie ihn:

```
invis:~ # systemctl start kopano-gateway.service
```

Auch dies sollten Sie überprüfen:

```
invis:~ # systemctl status kopano-gateway.service
```

Auch dabei ist bei unseren Tests nichts schief gegangen.

Gehen Sie identisch beim Kopano-ical Dienst vor:

```
invis:~ # old /etc/kopano/ical.cfg
moving /etc/kopano/ical.cfg to /etc/kopano/ical.cfg-20190829
invis:~ # cp /usr/share/sine/templates/groupware/kopano/ical.cfg
/etc/kopano/
```

```
invis:~ # systemctl start kopano-ical.service
```

Auch dies sollten Sie überprüfen:

```
invis:~ # systemctl status kopano-ical.service
```

Zu guter Letzt fehlt noch eine zusätzliche Konfigurationsdatei, die Sie ebenfalls aus unseren Vorlagen beziehen können:

```
invis:~ # cp /usr/share/sine/templates/groupware/kopano/admin.cfg
/etc/kopano/
```

Sollten Sie die kommerzielle Kopano-Erweiterung „Files“ installiert haben müssen Sie noch ein PHP-Paket austauschen:

```
invis:~ # zypper in php7-smbclient
```

Das zu ersetzende Paket wird dabei automatisch deinstalliert. Starten Sie anschließend den Apache-Webserver neu:

```
invis:~ # systemctl restart apache2.service
```

Sie können jetzt noch einmal alle Kopano-Dienste im Block, stoppen, starten und überprüfen:

```
invis:~ # runkopano stop
invis:~ # runkopano start
invis:~ # runkopano status
```

Laufen alle Dienst, ist auch das Kopano Upgrade abgeschlossen.

Kimai aktualisieren

Wenn Sie die Zeiterfassungssoftware Kimai auf dem invis-Server nutzen, muss dessen Datenbank an die während des Upgrades installierte neue Kimai-Version angepasst werden. Dieser Vorgang läuft völlig automatisch ab. Öffnen Sie Kimai einfach wie gewohnt und folgen Sie den Anweisungen.

ownCloud aktualisieren

Zur Aktualisierung von ownCloud (falls erforderlich) finden Sie hier im Wiki einen eigenen Artikel.

Normalerweise sollten aber folgende Kommandos ausreichen um ownCloud auf Stand zu bringen:

```
invis:~ # sudo -u wwwrun /srv/www/htdocs/owncloud/occ upgrade
...
invis:~ # sudo -u wwwrun /srv/www/htdocs/owncloud/occ maintenance:mode --off
```

Bleibt nur noch: Viel Spaß mit Ihrem invis-Server 14.1!

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

Permanent link:
https://wiki.invis-server.org/doku.php?id=invis_server_wiki:upgrade:13.5_to_14.1&rev=1589870449

Last update: **2020/05/19 06:40**