

# Mehrere Half-Life bzw. Half-Life 2 Gameserver auf elegantem Weg einrichten

**Dieses Konzept soll veranschaulichen, wie mehrere Gameserver mittels Symlinks eingerichtet werden können.**

Da sich immer mehr Clans einen dedizierten Linux-Server mieten (Root-Server), taucht immer öfters die Frage auf, wie man am besten mehrere Gameserver einrichtet.

Oft wird geraten einfach ein Webinterface zu nutzen, doch nicht jeder möchte sich aber so ein Webinterface auf seinem Server einrichten.

Der simpelste Weg ist es die Server einfach zu kopieren oder getrennt voneinander mit dem Programm steam herunterzuladen bzw. zu installieren. Spätestens bei einem Steamupdate kommt dann das böse Erwachen. Zu oft kommt es vor, dass die Content-Server von Valve total überlastet sind und die Updates schneller an die Clients verteilt werden, als an die Server. Der große Nachteil bei mehreren Servern ist deswegen das Updaten. Jeder Server muss einzeln aktualisiert werden. Das nimmt viel Zeit in Anspruch und oft können die Spieler nur noch auf aktuellen Servern spielen. Meistens klappt dann das Update bei einem Server, aber bei den restlichen nicht. Wenn man einen schlechten Zeitpunkt erwischt, ist man machtlos. Zum Glück hat sich die Lage etwas verbessert.

Eine Möglichkeit besteht darin einen Hauptserver zu aktualisieren und dieses Update mittels 'cp -uR' (kopiert nur neuere oder fehlende Dateien) auf die anderen Server zu kopieren. Mit Tricks kann man verhindern, dass Dateien wie z.B. die maplist.txt, motd.txt oder mapcycle.txt nicht überschrieben werden.

Der eleganteste Weg ist Symlinks zu nutzen. Die Grundidee ist nur einen Hauptserver einzurichten, der die ständig aktualisiert wird. Die anderen Server nutzen die Dateien vom Hauptserver. In der Verzeichnisstruktur sind die Server der des Hauptservers identisch, sind aber in ihrem eignen Verzeichnis untergebracht. Die Dateien vom Hauptserver werden verlinkt.

Es gibt 2 Arten von Links.

**Hardlink:** Ein harter Link (engl. hard link) ist ein Verzeichniseintrag in einem Dateisystem (z.B. ext3). Dateien und Verzeichnisse werden dort systemintern über die Nummer ihres Dateideskriptors, des Inodes, identifiziert; insbesondere wenn mehrere Verzeichniseinträge auf denselben Inode verweisen, spricht man von „harten Links“. Im Unterschied hierzu verweisen symbolische Links nicht auf einen Inode, sondern auf einen Verzeichniseintrag. Abgesehen von den Verzeichniseinträgen . und .. (für das aktuelle und das übergeordnete Verzeichnis) verweisen harte Links fast immer auf Dateien. Quelle: [wikipedia.de](http://wikipedia.de)

**Symlink:** Eine symbolische Verknüpfung, auch symbolischer Link, Symlink oder (fälschlicherweise) Softlink genannt, ist eine Verknüpfung in einem Dateisystem (Datei oder Verzeichnis), die auf eine andere Datei oder ein anderes Verzeichnis verweist. Es ist also lediglich eine Referenz und kein richtiges Element. Um einen Eintrag in das Dateiverzeichnis zu erstellen werden harte Links (engl. hardlink) verwendet. Quelle: [wikipedia.de](http://wikipedia.de)

Ich gehe auf die Möglichkeit ein mit Symlinks zu arbeiten, da sie einfacher zu handhaben sind.

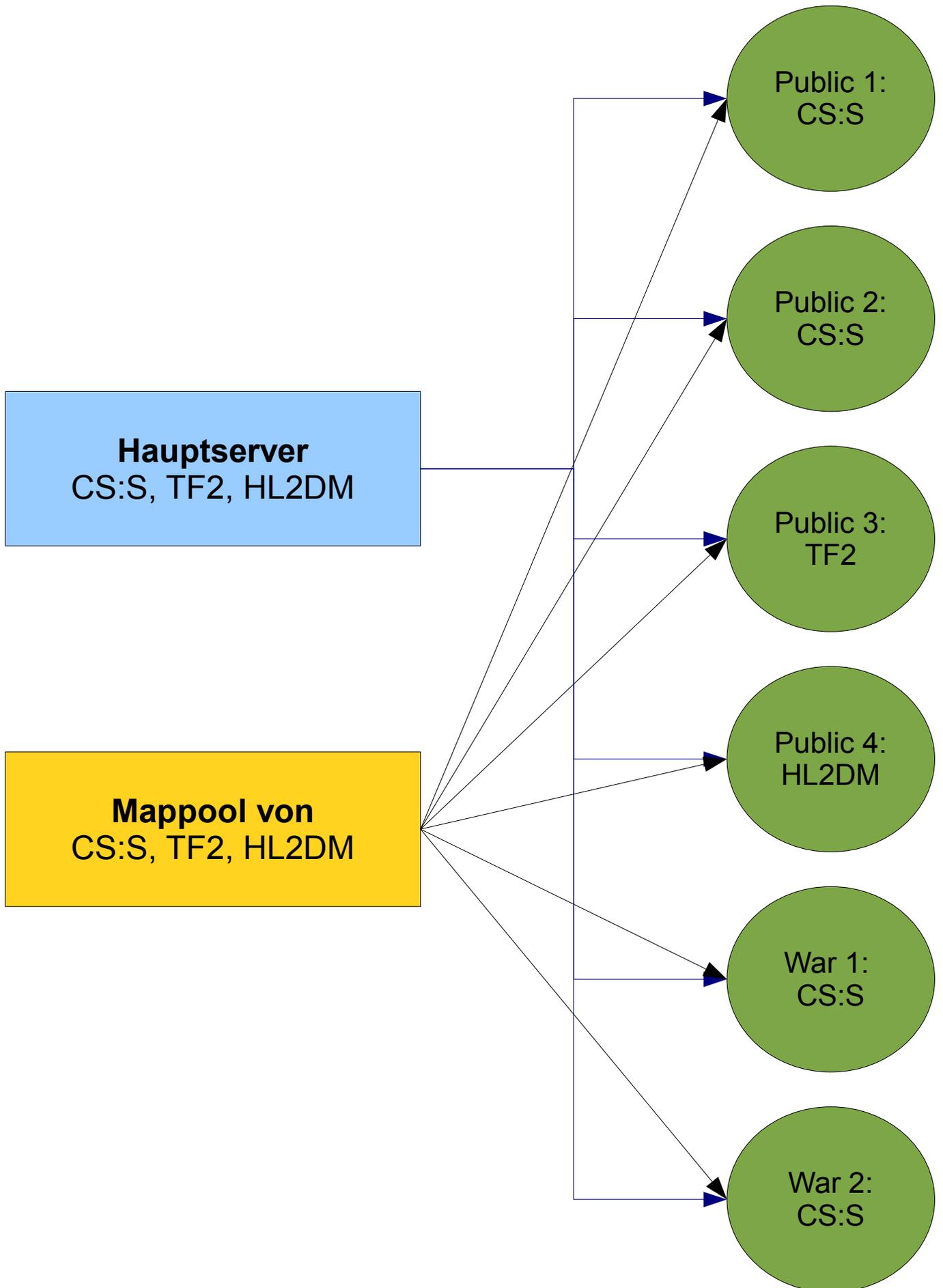
Um einzelne Dateien und Verzeichnisse zu verlinken, wird der Befehl ln verwendet. Für mehrere Dateien kann man den Befehl cp (copy) nutzen. In Verbindung der Optionsschalter -s (Symlink) und -R (Rekursiv), werden Dateien verlinkt und Verzeichnisse erstellt.

## **Folgende Vorteile ergeben sich durch das Nutzen von Symbolischen Links:**

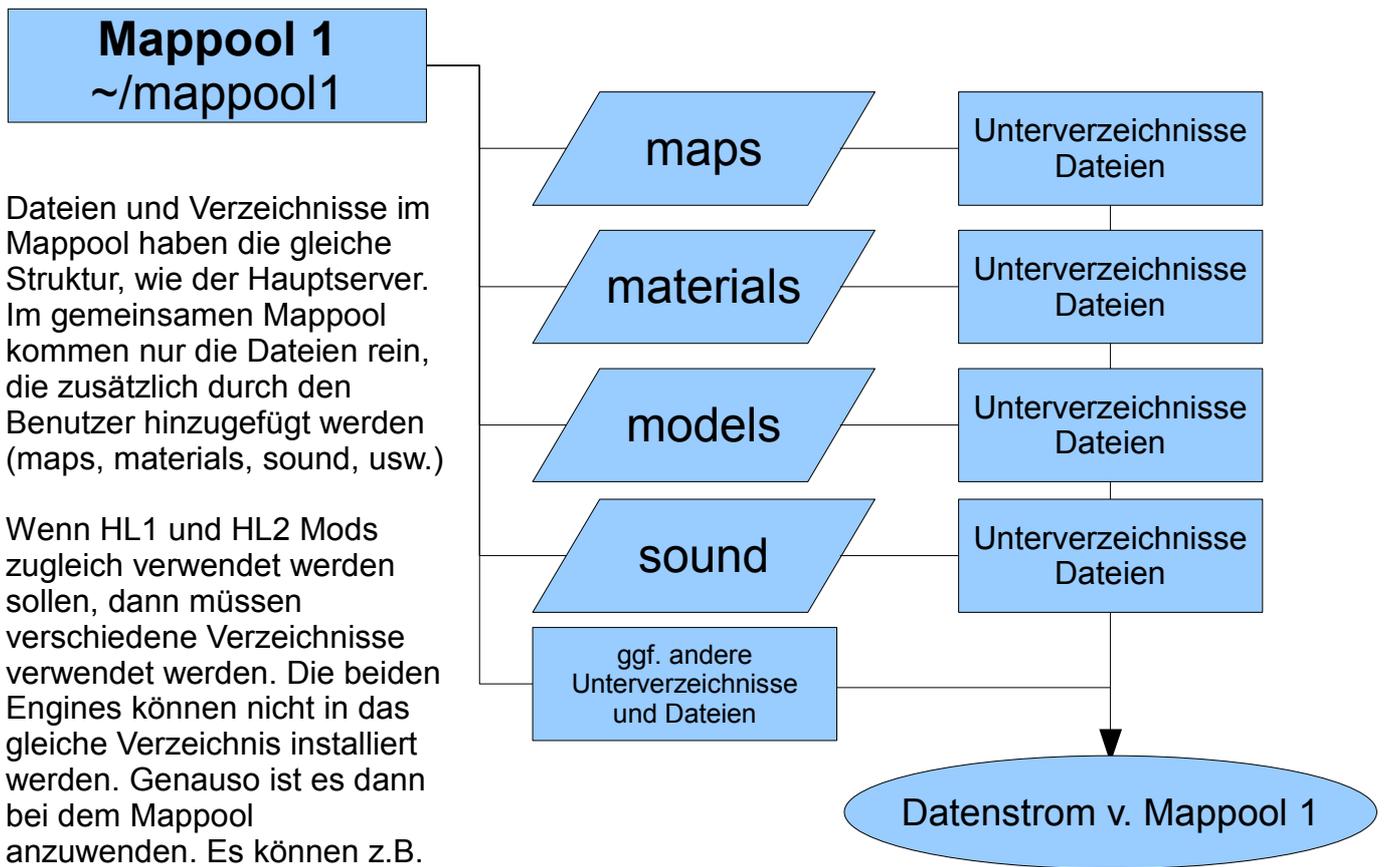
- Es muss nur ein Server mittels steam aktuell gehalten werden
- Neue Server lassen sich in weniger als 2 Minuten einrichten
- Ein netter Nebeneffekt ist der geringe Platzbedarf
- Es lässt sich sehr schnell ein Backup der Server machen
- Mappool kann unabhängig vom Hauptserver sein
- Schnelle Updates aller Server möglich

Fast alle Provider nutzen dieses Prinzip, weil es sonst nicht mehr anders Möglich wäre die Massen an Kundenservern aktuell zu halten.

# Vereinfachtes Beispiel

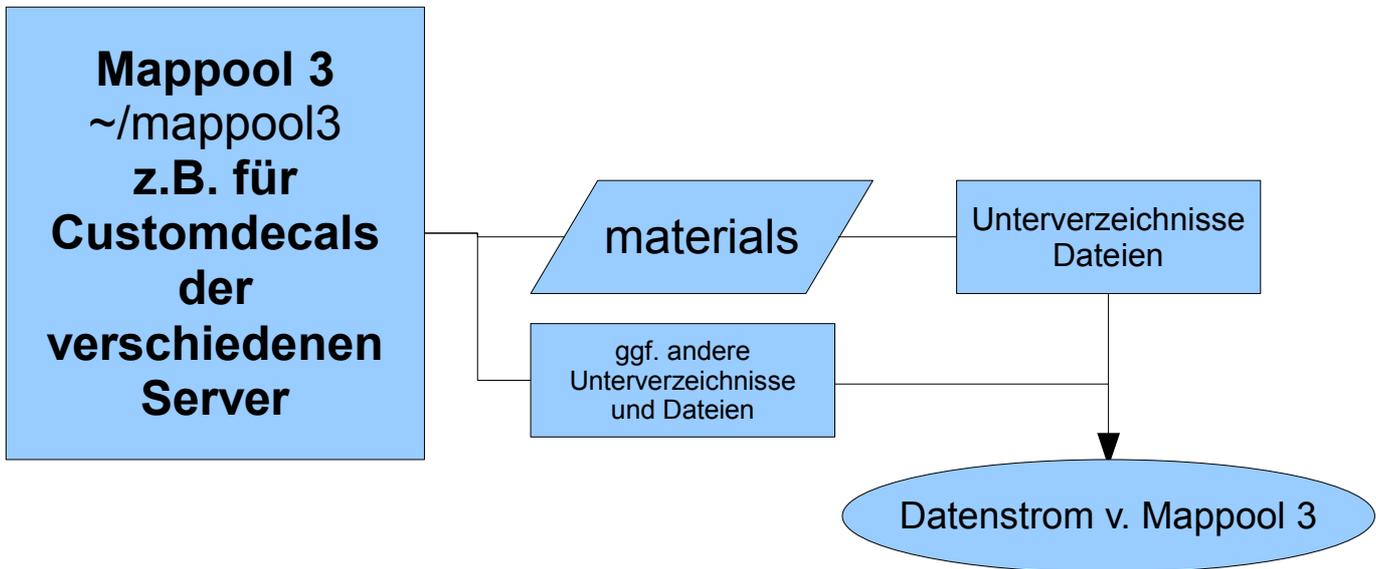
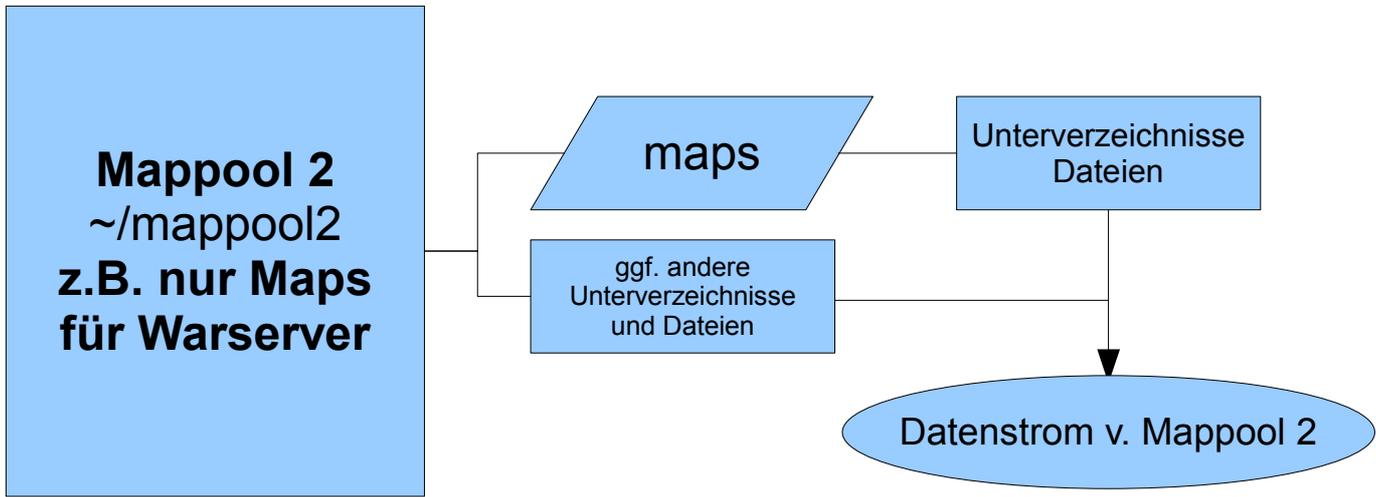


# Konzept mit verschiedenen Standard-Serverconfigs und mehreren Mappools



Dateien und Verzeichnisse im Mappool haben die gleiche Struktur, wie der Hauptserver. Im gemeinsamen Mappool kommen nur die Dateien rein, die zusätzlich durch den Benutzer hinzugefügt werden (maps, materials, sound, usw.)

Wenn HL1 und HL2 Mods zugleich verwendet werden sollen, dann müssen verschiedene Verzeichnisse verwendet werden. Die beiden Engines können nicht in das gleiche Verzeichnis installiert werden. Genauso ist es dann bei dem Mappool anzuwenden. Es können z.B. unterhalb des Hauptservers und des Mappools die Verzeichnisse hl1 und hl2 für die beiden Engines verwendet werden.



## Beispiel: Struktur des Mappools

```
mc - h1384381:/home/server/valve
server@h1384381:~/mappool_test$ ls -R
.:
h12  tesfile1  tesfile2

./h12:
cstrike

./h12/cstrike:
maps  materials  models  sound

./h12/cstrike/maps:
1701_SurfNcamp.bsp  Surf_BigBrigeFinal.nav
2teleporters.bsp   Surf_Combine1.bsp
King_Kong.nav       Surf_Crazytexture.bsp
Office_Space.bsp   Surf_Crazytexture.nav

./h12/cstrike/materials:
420  Alamo

./h12/cstrike/materials/420:
420cent001.vtf

./h12/cstrike/materials/Alamo:
mdstomb.tga  mdstomb.txt  mdstomb.vtf

./h12/cstrike/models:
de_forge

./h12/cstrike/models/de_forge:
forge_fern1.mdl

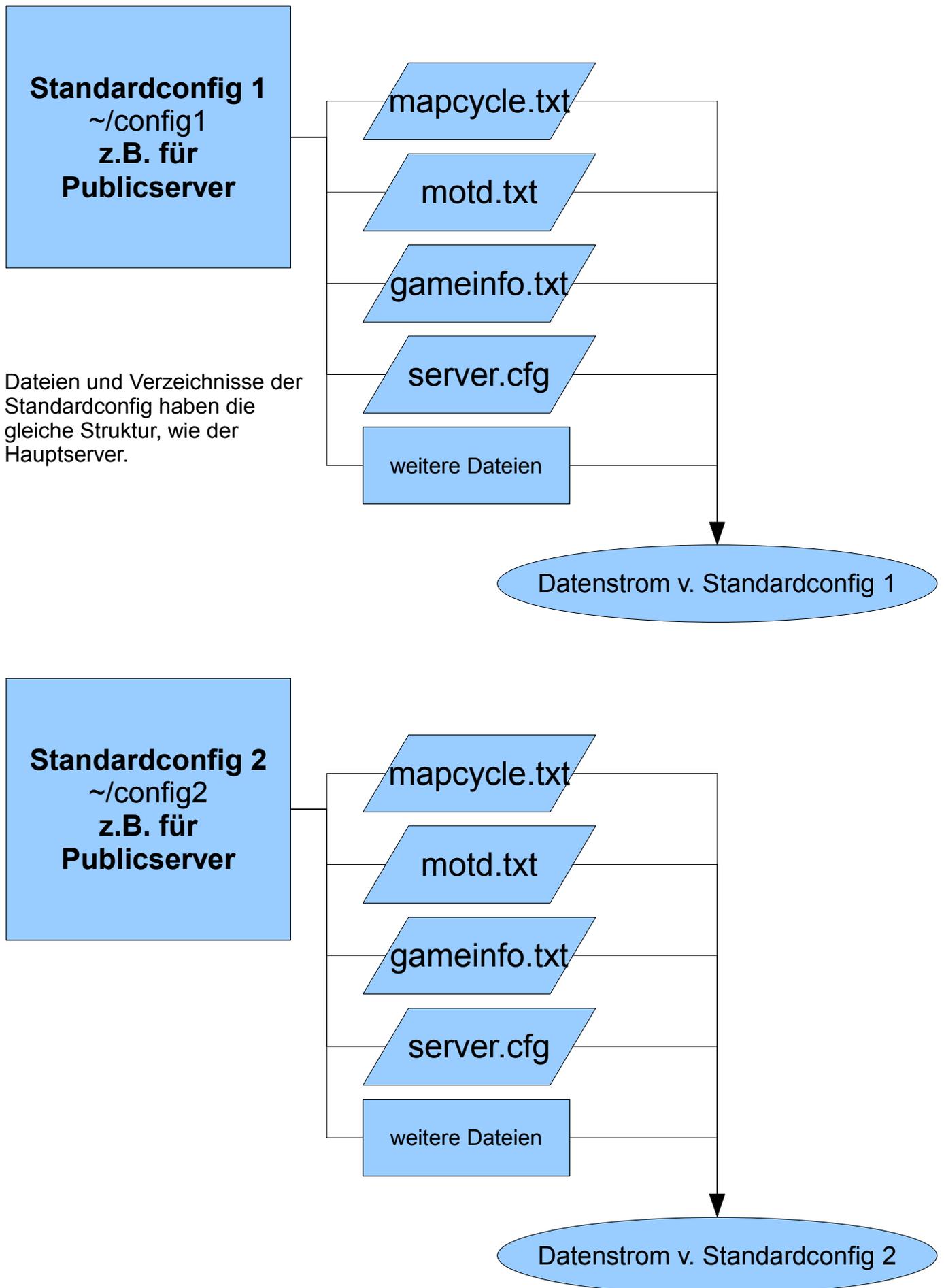
./h12/cstrike/sound:
747  aim_pasture

./h12/cstrike/sound/747:
mechwhine.wav  plane_drone.wav

./h12/cstrike/sound/aim_pasture:
server@h1384381:~/mappool_test$
```

# Beispiel: Standardconfigs für Server erstellen

Beim installieren, werden diese Cofigs dann kopiert.



## Beispiel: Struktur der Standard-Serverconfig

```
mc - h1384381:/home/server/valve
server@h1384381:~/valve/valveconfig$ ls -R
.:
h11  h12

./h11:
cstrike  dod

./h11/cstrike:
liblist.gam  mapcycle.txt  motd.txt  server.cfg

./h11/dod:
autoexec.cfg  mapcycle.txt  server.cfg
liblist.gam  motd.txt

./h12:
cstrike  hl2mp

./h12/cstrike:
addons  cfg  gameinfo.txt  mapcycle.txt  motd.txt

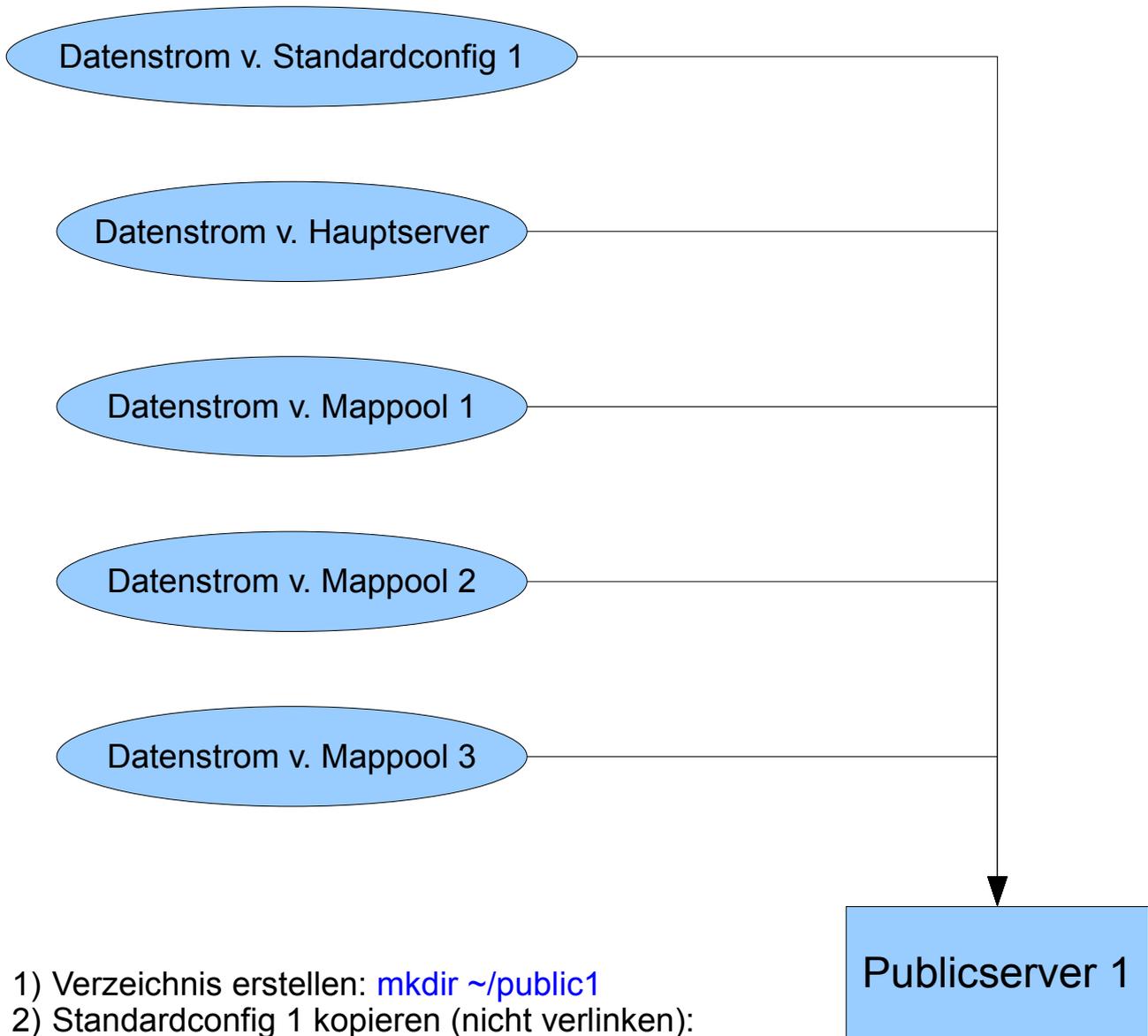
./h12/cstrike/addons:

./h12/cstrike/cfg:
autoexec.cfg  server.cfg

./h12/hl2mp:
cfg  gameinfo.txt  mapcycle.txt  motd.txt

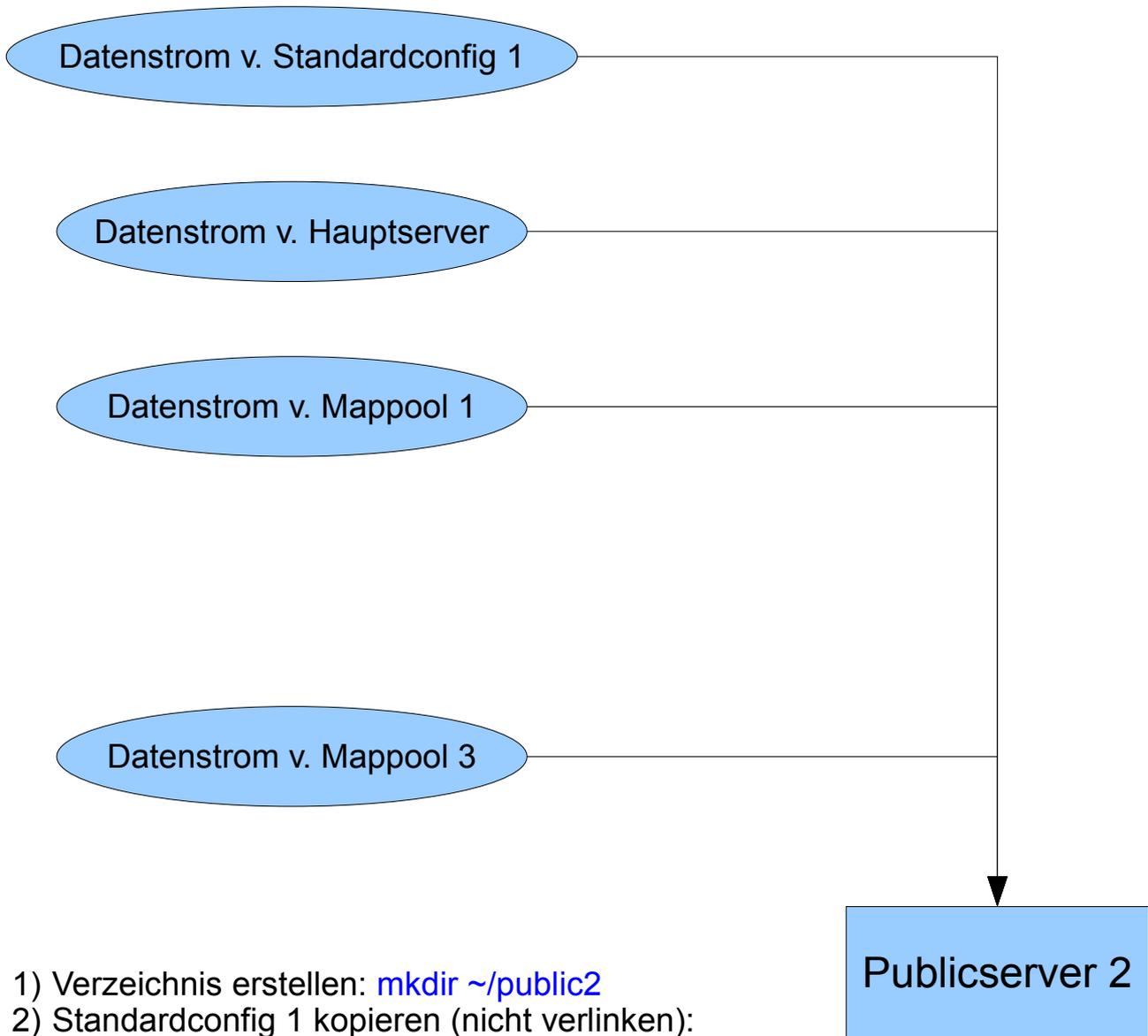
./h12/hl2mp/cfg:
server@h1384381:~/valve/valveconfig$
```

# Erstellen eines Publicservers



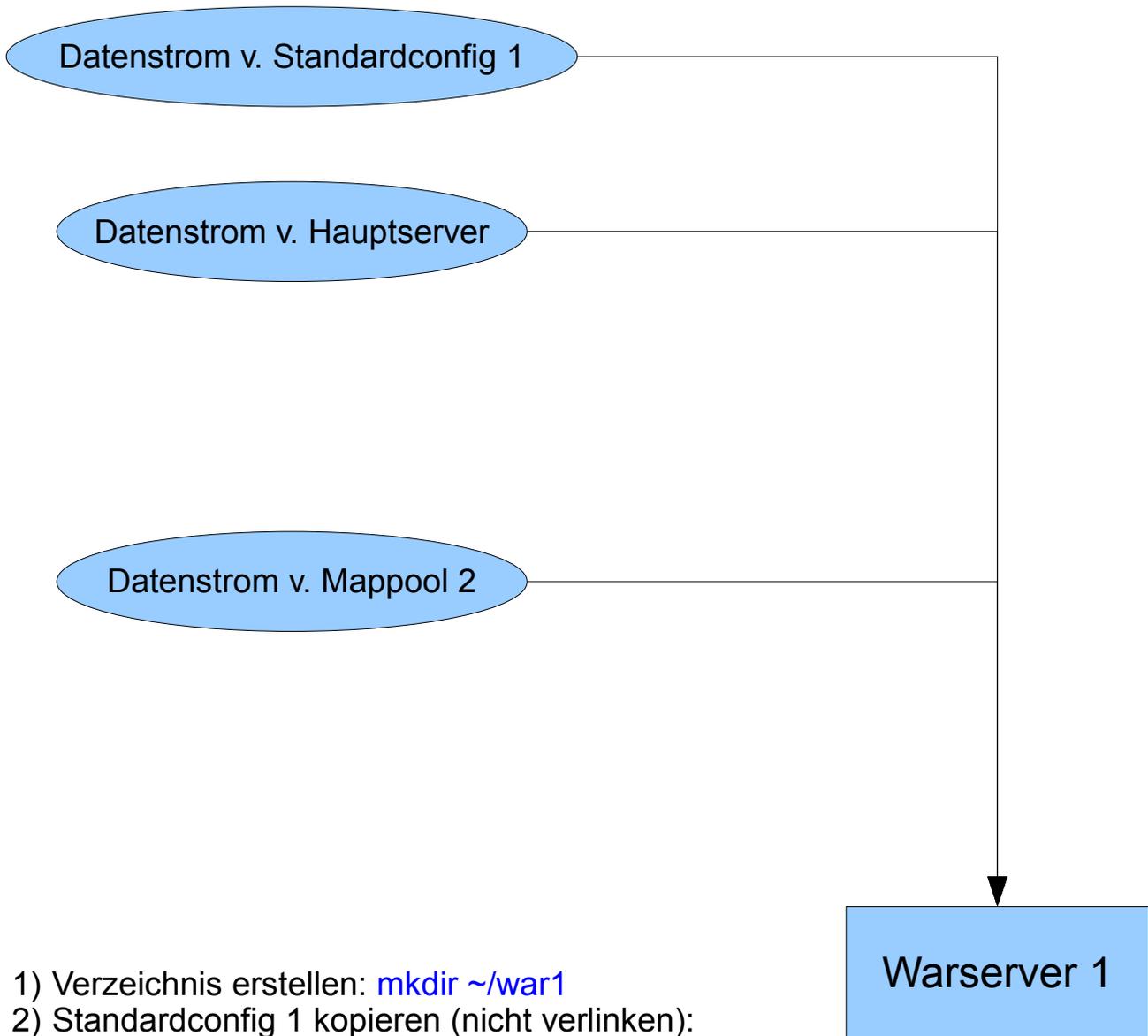
- 1) Verzeichnis erstellen: `mkdir ~/public1`
- 2) Standardconfig 1 kopieren (nicht verlinken):  
`cp -R ~/config1/* ~/public1/`
- 3) Dateien vom Hauptserver verlinken:  
`cp -sR ~/hauptserver/* ~/public1/`
- 4) Dateien vom Mappool 1 verlinken:  
`cp -sR ~/mappool1/* ~/public1/`
- 5) Mappool 2: `cp -sR ~/mappool2/* ~/public1/`
- 6) Mappool 3: `cp -sR ~/mappool3/* ~/public1/`

# Erstellen eines anderen Publicservers



- 1) Verzeichnis erstellen: `mkdir ~/public2`
- 2) Standardconfig 1 kopieren (nicht verlinken):  
`cp -R ~/config1/* ~/public2/`
- 3) Dateien vom Hauptserver verlinken:  
`cp -sR ~/hauptserver/* ~/public2/`
- 4) Dateien vom Mappool 1 verlinken:  
`cp -sR ~/mappool1/* ~/public2/`
- 5) Mappool 3: `cp -sR ~/mappool3/* ~/public2/`

# Erstellen eines Warservers



- 1) Verzeichnis erstellen: `mkdir ~/war1`
- 2) Standardconfig 1 kopieren (nicht verlinken):  
`cp -R ~/config2/* ~/war1/`
- 3) Dateien vom Hauptserver verlinken:  
`cp -sR ~/hauptserver/* ~/war1/`
- 4) Dateien vom Mappool 2 verlinken:  
`cp -sR ~/mappool2/* ~/war1/`

# Update aller Server nach einem Steamupdate von mehreren Mods (CS1.6, DOD, DOD:S, HL2DM, TF2...)

## Steamupdate

```
./steam -command update -game tf -dir ~/hauptserver/hl2  
./steam -command update -game hl2mp -dir ~/hauptserver/hl2  
./steam -command update -game dods -dir ~/hauptserver/hl2  
./steam -command update -game "Counter-Strike Source" -dir ~/hauptserver/hl2  
./steam -command update -game cstrike -dir ~/hauptserver/hl1  
./steam -command update -game valve -dir ~/hauptserver/hl1  
./steam -command update -game dod -dir ~/hauptserver/hl1
```

## Symlinks der Server aktualisieren

```
cp -sR ~/hauptserver/* ~/public1/  
cp -sR ~/hauptserver/* ~/public2/  
cp -sR ~/hauptserver/* ~/war1/
```

**Alle Server wieder aktuell**

**Neue Maps sind zum gemeinsamen  
Mappool hinzugefügt worden.  
Die Symlinks müssen danach für die einzelnen  
Server erstellt werden.**

### **Public 1**

```
cp -sR ~/mappool1/* ~/public1/  
cp -sR ~/mappool2/* ~/public1/  
cp -sR ~/mappool3/* ~/public1/
```



### **Public 2**

```
cp -sR ~/mappool1/* ~/public2/  
cp -sR ~/mappool3/* ~/public2/
```



### **Warserver 1**

```
cp -sR ~/mappool2/* ~/war1/
```