

ARK-Cluster Einrichtung (Cross-ARK) [3 Varianten | Linux]

Inhaltsverzeichnis

- [1 Vorbereitung](#)
- [2 Variante 1 - mehrere ARK-Server im selben Server Verzeichnis](#)
- [3 Variante 2 - mehrere ARK-Server in unterschiedlichen Server Verzeichnissen](#)
- [4 Variante 3 - mehrere ARK-Server auf mehreren V- oder Rootservern \(Cross ARK über die Cloud\)](#)

Als Betriebssystem für dieses Tutorial haben wir Debian 8 genutzt.

1 Vorbereitung

Diese schritte sind bitte bei allen Varianten zu befolgen.
Alle ARK Server sollten bereits installiert sein.

- Fügt eurem Startbefehl/Startscript folgende Parameter hinzu:
 - `-clusterid=clusterxyz` das "clusterxyz" bitte durch etwas eigenes einzigartiges ersetzen. Die Cluster ID muss für alle Server im Cluster gleich sein.
 - optional [EMPFOHLEN] `-NoTransferFromFiltering` Dieser Parameter verhindert, dass [Spieler](#) ihre Dinos, Items und Charakter in den Singelplayer mitnehmen können und oder ihre Dinos, Items und Charakter aus dem Singelplayer mit in den Cluster mitbringen können, Kurzform: Down- u. Uploades sind nur innerhalb der Server welche sich im Cluster befinden möglich.
- Fügt nachfolgende Einstellungen in eure `GameUserSettings.ini` ein:
- `PreventDownloadSurvivors=False` erlaubt (False) oder verbietet (True) den Download des Charakters auf dem Server.
- `PreventDownloadItems=False` erlaubt (False) oder verbietet (True) den Download von Items auf dem Server.
- `PreventDownloadDinos=False` erlaubt (False) oder verbietet (True) den Download von Dinos auf dem Server.
- `PreventUploadSurvivors=False` erlaubt (False) oder verbietet (True) den Upload des Charakters auf dem Server.
- `PreventUploadItems=False` erlaubt (False) oder verbietet (True) den Upload von Items auf dem Server.
- `PreventUploadDinos=False` erlaubt (False) oder verbietet (True) den Upload von Dinos auf dem Server.

2 Variante 1 - mehrere ARK-Server im selben Server Verzeichnis

In dieser Variante werden mehrere ARK-Server im selben Verzeichnis mit unterschiedlichen Startbefehlen gestartet.

Fügt eurem Startbefehl/Startscript zusätzlich folgende Parameter hinzu:

- `?AltSaveDirectoryName=xyz` das "xyz" bitte gegen etwas eigenes ersetzen, für jeden Server sollte dort was anderes eingetragen werden. Dieser Parameter erstellt für jeden Server einen extra Ordner mit den Save Files.

Nachfolgend ein Beispiel für 2 Startbefehle:

In diesem Beispiel liegen beide ARK-Server im Verzeichnis `/home/arkserver`.

Beide Startbefehle werden im Verzeichnis `/home/arkserver/ShooterGame/Binaries/Linux` ausgeführt.

```
Server1:                                     ./ShooterGameServer
TheIsland?MaxPlayers=50?Port=27015?AltSaveDirectoryName=Server1?listen           -
NoTransferFromFiltering -clusterid=clsuter123 -server -log
Server2:                                     ./ShooterGameServer
TheCenter?MaxPlayers=50?Port=27025?AltSaveDirectoryName=Server2?listen           -
NoTransferFromFiltering -clusterid=clsuter123 -server -log
```

Achtet bitte darauf, dass ihr für jeden Server einen anderen Port wählt.

Der Nachteil bei dieser Variante ist, alle Server nutzen die selbe `GameUserSettings.ini` und die selbe `Game.ini`

Alle Server müssen auf dem Selben vServer bzw. Rootserver liegen.

3 Variante 2 - mehrere ARK-Server in unterschiedlichen Server Verzeichnissen

In dieser Variante werden mehrere ARK-Server in unterschiedlichen Server Verzeichnissen gestartet.

- Legt irgendwo auf eurem vServer oder Root Server ein Verzeichnis an, auf welches alle ARK-Server Zugriff haben.
-

Fügt eurem Startbefehl/Startscript zusätzlich folgende Parameter hinzu:

- `-ClusterDirOverride=/home/ark-cluster "/home/ark-cluster"` bitte in den Pfad zum eben erstellten Verzeichnis ändern.

Nachfolgend ein Beispiel für 2 Startbefehle:

In diesem Beispiel liegt Server1 im Verzeichnis `/home/server1` und Server2 im Verzeichnis `/home/server2`.

Unser Cluster Verzeichnis liegt `/home/ark-cluster`

Startbefehl für Server1 wird im Verzeichnis `/home/server1/ShooterGame/Binaries/Linux` ausgeführt.

Startbefehl für Server2 wird im Verzeichnis `/home/server2/ShooterGame/Binaries/Linux` ausgeführt.

```
Server1:    ./ShooterGameServer      TheIsland?MaxPlayers=50?Port=27015?listen    -
ClusterDirOverride=/home/ark-cluster          -NoTransferFromFiltering                    -
clusterid=clsuter123 -server -log
Server2:    ./ShooterGameServer      TheCenter?MaxPlayers=50?Port=27025?listen    -
ClusterDirOverride=/home/ark-cluster          -NoTransferFromFiltering                    -
clusterid=clsuter123 -server -log
```

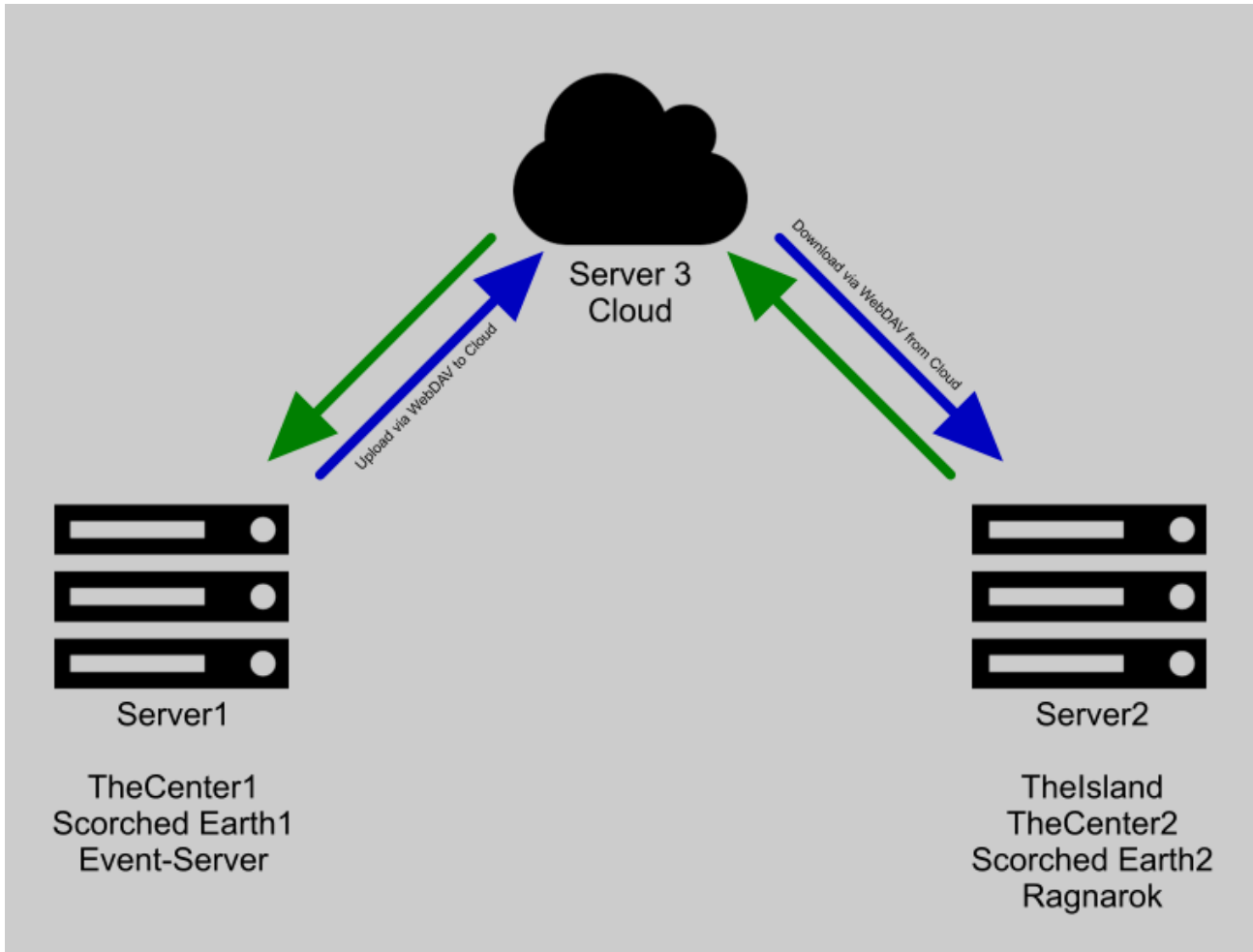
Achtet bitte darauf, dass ihr für jeden Server einen anderen Port wählt.
Alle Server müssen auf dem Selben vServer bzw. Rootserver liegen.

4 Variante 3 - mehrere ARK-Server auf mehreren V- oder Rootservern (Cross ARK über die Cloud)

In dieser Variante werden mehrere ARK-Server auf unterschiedlichen Hostsystemen gestartet und über einen WebDAV zugang verbunden.

Für unseren Cluster nutzen wir 3 Rootserver, auf Root1 laufen 3 ARKs auf Root2 laufen 4 ARKs und auf Root3 läuft ein Cloudsystem mit WebDAV Unterstützung.

Folgende Grafik veranschaulicht das ganze:



Es ist nicht unbedingt eine eigene Cloud notwendig, diverse Cloudanbieter (z.b. OneDrive von Microsoft) unterstützen ebenfalls eine WebDAV Zugang an. Ggf kann auch ein 100 MB Account in unserer Cloud zur Verfügung gestellt werden, hierzu bitte eine PN an mich.

- Legt irgendwo auf eurem vServer oder Root Server ein Verzeichnis an, auf welches alle ARK-Server Zugriff haben.
- Fügt eurem Startbefehl/Startscript zusätzlich folgende Parameter hinzu:
 - `-ClusterDirOverride=/home/ark-cluster "/home/ark-cluster"` bitte in den Pfad zum eben erstellten Verzeichnis ändern.
- Als root via SSH einloggen.
- folgenden Befehl `apt-get install davfs2 fuse` ausführen.
 - Während der Installation wird ein blauer Bildschirm kommen wo gefragt wird ob User eine WebDAV Verbindung nutzen dürfen. Diese Frage mit **Ja** oder **Yes** beantworten!
- Jetzt die Datei "fstab" mit `nano /etc/fstab` öffnen und unten folgendes einfügen:

```
◦ https://WebDAV-Link.de/irgendwas /home/ark-cluster davfs
  user,rw,noauto 0 0
```

anschließend die Datei speichern und schließen.

- Jetzt den User auf welchen die ARK-Server laufen zur Gruppe "davfs2" mit diesen Befehl hinzufügen:
`sudo usermod -a -G davfs2 XXXXX`
 - Die XXXXX sind bitte durch den User auf welchen die ARK-Server laufen zu ersetzen.
- Jetzt via SSH mit dem User auf welchen die ARK-Server laufen anmelden (`su XXXXX`)
- Jetzt folgenden Befehl eingeben: `mount /home/ark-cluster`
- Jetzt wird nach einem Benutzernamen und einem Passwort gefragt, dies sind die selben Login-Daten mit der sich in den Cloud Account eingeloggt wird.

Diese Schritte sind auf allen v- und oder Rootservern zu wiederholen.

Verbindung zur Cloud trennen (falls notwendig): `fusermount -u /home/ark-cluster`

Nachfolgend ein Beispiel für 2 Startbefehle:

In diesem Beispiel liegt **Server1** im Verzeichnis `/home/arkserver` auf Root1 und **Server2** im Verzeichnis `/home/arkserver` auf Root2.

Unser Cluster Verzeichnis liegt `/home/ark-cluster`

Startbefehl für Server1 wird im Verzeichnis `/home/arkserver/ShooterGame/Binaries/Linux` auf Root1 ausgeführt.

Startbefehl für Server2 wird im Verzeichnis `/home/arkserver/ShooterGame/Binaries/Linux` auf Root2 ausgeführt.

```
Server1:    ./ShooterGameServer      TheIsland?MaxPlayers=50?Port=27015?listen  -  
ClusterDirOverride=/home/ark-cluster        -NoTransferFromFiltering  -  
clusterid=clsuter123 -server -log  
Server2:    ./ShooterGameServer      TheCenter?MaxPlayers=50?Port=27015?listen  -  
ClusterDirOverride=/home/ark-cluster        -NoTransferFromFiltering  -  
clusterid=clsuter123 -server -log
```