

# Usenet Download Server

## Basic System Setup

```
# curl -sSL
https://www.blackgate.org/deployment/base_configuration_blackGATE.sh | bash
# yum update && yum install nfs-utils yum-utils epel-release
# systemctl start rpcbind nfslock && systemctl enable rpcbind nfslock

# groupadd usenetserver
# mkdir /mnt/blackserv
# chgrp -R usenetserver /mnt/blackserv/ && chmod -R 775 /mnt/blackserv/
# vim /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Wed Dec 6 13:07:04 2017
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos_rhdl03-root / xfs defaults
0 0
UUID=a34686fa-4b05-43ed-b3bb-c1d754783bc2 /boot xfs
defaults 0 0
UUID=D653-2F5A /boot/efi vfat
umask=0077,shortname=winnt 0 0
/dev/mapper/centos_rhdl03-opt /opt xfs defaults
0 0
/dev/mapper/centos_rhdl03-swap swap swap defaults
0 0

10.0.0.21:/volume1/media /mnt/blackserv nfs rw,hard,intr 0 0
```

```
# mount -a
```

## Webserver / Application Proxy Setup

```
# yum -y install https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
# yum update
# yum install -y wget git tar vim curl policycoreutils-python httpd php71w
php71w-dom php71w-mbstring php71w-gd php71w-json \
php71w-xml php71w-zip php71w-curl php71w-mcrypt php71w-pear php71w-opcache
```

```
# sed -i 's/^\#&/g' /etc/httpd/conf.d/welcome.conf
# sed -i 's/DocumentRoot "\/var\www\html"/DocumentRoot
"\/var\www\html\muximum"/' /etc/httpd/conf/httpd.conf

# mkdir /var/www/html/muximum
# usermod -a -G apache michael
# git clone https://github.com/mescon/Muximum.git /var/www/html/muximum/
# chown -R apache:apache /var/www/html/muximum/
# chmod -R 775 /var/www/html/muximum/
# ls -l /var/www/html/muximum/

# vim /etc/httpd/conf.d/usenet_proxy.conf
```

```
ServerName www.blackgate.org
ServerTokens Prod
ServerSignature Off

<Directory /var/www/html/muximum>
    Options +FollowSymlinks
    AllowOverride All
    # Allow open access:
    Require all granted
</Directory>

<Location /nzbget/>
    ProxyPass http://127.0.0.1:6789/
    ProxyPassReverse http://127.0.0.1:6789/
</Location>

<Location /nzbhydra/>
    ProxyPass http://127.0.0.1:5075/
    ProxyPassReverse http://127.0.0.1:5075/
</Location>

<Location /sonarr/>
    RewriteEngine On
    RewriteRule /sonarr/(.*)
http://127.0.0.1:8989/usenetdl/sonarr/$1 [P]
    ProxyPassReverse http://127.0.0.1:8989/
</Location>

<Location /radarr/>
    ProxyPassReverse http://127.0.0.1:7878/
    RewriteEngine On
    RewriteRule /radarr/(.*)
http://127.0.0.1:7878/usenetdl/radarr/$1 [P]
</Location>

<Location /lidarr/>
    ProxyPassReverse http://127.0.0.1:8686/
    RewriteEngine On
```

```
RewriteRule      /lidarr/(.*)
http://127.0.0.1:8686/usenetdl/lidarr/$1 [P]
</Location>
```

```
Header set X-Content-Type-Options: "nosniff"
Header set X-Frame-Options: "sameorigin"
```

```
# setsebool httpd_can_network_connect 1 -P
# setsebool -P httpd_execmem 1
# semanage fcontext -a -t httpd_sys_rw_content_t
'/var/www/html/muximux(/.*)?'
# semanage fcontext -a -t httpd_sys_rw_content_t '/var/www/html/muximux'
# restorecon -Rv /var/www/html/
# ls -lZ /var/www/html/

# systemctl start httpd && systemctl status httpd
# systemctl enable httpd

# firewall-cmd --permanent --zone=public --add-service=http
# firewall-cmd --permanent --zone=public --add-service=https
# firewall-cmd --reload
```

**Achtung:** Für dieses Setup werden zwei Reverse Proxies gebraucht. Dieser Proxy, den wir gerade eingerichtet haben, proxyt ausschliesslich die Anfragen von zweiter Stelle (den ServiceNamen) auf das Backend. Zum proxen aller Services unter dem Sub-Verzeichniss “/usenetdl” wird der haupt Reverse Proxy (mit TLS-Termination) benötigt

Beispiel Haupt-Proxy Stanza:

```
...
RewriteEngine on
RewriteRule ^/usenetdl$ /usenetdl/ [R]

ProxyPass      /usenetdl/ http://172.168.0.11/
ProxyPassReverse /usenetdl/ http://172.168.0.11/
```

## Install Usenet Services

### Setup NZBget - Binary Downloader

```
# cd /opt/
# wget https://nzbget.net/download/nzbget-latest-bin-linux.run
# sh nzbget-latest-bin-linux.run
# ls -l
# cd /opt/nzbget/
# ./nzbget -s
```

```
# useradd -r nzbget
# usermod -a -G usenetserver nzbget
# chown -R nzbget:usenetserver /opt/nzbget

# vim /etc/systemd/system/nzbget.service
```

```
[Unit]
Description=NZBGet Daemon
Documentation=http://nzbget.net/Documentation
After=network.target

[Service]
User=nzbget
Group=usenetserver
Type=forking
ExecStart=/opt/nzbget/nzbget -c /opt/nzbget/nzbget.conf -D
ExecStop=/opt/nzbget/nzbget -Q
ExecReload=/opt/nzbget/nzbget -O
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

```
# systemctl daemon-reload
# systemctl start nzbget.service && systemctl status nzbget.service
# systemctl enable nzbget.service

# vim /opt/nzbget/nzbget.conf
```

```
...
MainDir=${AppDir}/downloads
...
DestDir=/mnt/blackserv/downloads/_done
...
InterDir=/mnt/blackserv/downloads/intermediate
...
NzbDir=/mnt/blackserv/downloads/nzb
...
AuthorizedIP=127.0.0.1
...
```

Der Rest von NZBget wird via Web-GUI konfiguriert

## Setup NzbHydra - Usenet Search Tool

```
# yum groupinstall -y "development tools"
# yum install -y zlib-devel openssl-devel sliteq-devel bzip2-devel python
ncurses-devel sqlite-devel readline-devel tk-devel gdbm-devel db4-devel
libpcap-devel xz-devel expat-devel

# cd /tmp/
# wget http://python.org/ftp/python/2.7.14/Python-2.7.14.tar.xz
# tar xf Python-2.7.14.tar.xz
# cd Python-2.7.14
# ./configure --prefix=/usr/local --enable-unicode=ucs4 --enable-shared
LDFLAGS="-Wl,-rpath /usr/local/lib"
# make && make altinstall
# /usr/local/bin/python2.7

# git clone https://github.com/theotherp/nzbhydra /opt/nzbhydra

# useradd -r nzbhydra
# usermod -a -G usenetserver nzbhydra
# chown -R nzbhydra:usenetserver /opt/nzbhydra
# ls -l /opt/nzbhydra/

# sed -e 's/"urlBase": null,/"urlBase": "\/usenetdl\/nzbhydra",/' -i
/opt/nzbhydra/settings.cfg
# /usr/local/bin/python2.7 /opt/nzbhydra/nzbhydra.py --daemon --nobrowser
# cat /opt/nzbhydra/settings.cfg | grep /nzbhydra
```

```
"externalUrl": "https://blackgate.org/usenetdl/nzbhydra",
"urlBase": "\/usenetdl\/nzbhydra",
```

```
# vim /etc/systemd/system/nzbhydra.service
```

```
[Unit]
Description=NZBHydra Daemon
Documentation=https://github.com/theotherp/nzbhydra
After=network.target

[Service]
User=nzbhydra
Group=usenetserver
Type=simple
ExecStart=/usr/local/bin/python2.7 /opt/nzbhydra/nzbhydra.py --nobrowser
```

```
KillMode=process
Restart=on-failure
```

```
[Install]
WantedBy=multi-user.target
```

```
# systemctl daemon-reload
# systemctl start nzbhydra.service && systemctl status nzbhydra.service
# systemctl enable nzbhydra.service

# netstat -tulpen
```

---

## Setup Sonarr - TV Download Tool

```
# rpm --import
"http://keyserver.ubuntu.com/pks/lookup?op=get&search=0x3FA7E0328081BFF6A14D
A29AA6A19B38D3D831EF"
# yum-config-manager --add-repo
http://download.mono-project.com/repo/centos/
# yum install -y mediainfo libzen libmediainfo curl gettext mono-core mono-
devel sqlite mono-locale-extras wget git par2cmdline p7zip unrar unzip tar
gcc python-feedparser python-configobj python-cheetah python-dbus python-
devel libxslt-devel

# cd /tmp/
# useradd -r sonarr
# usermod -a -G usenetserver sonarr
# wget http://update.sonarr.tv/v2/master/mono/NzbDrone.master.tar.gz
# tar -xvf NzbDrone.master.tar.gz
# mkdir -p /opt/sonarr/bin

# cp -a /tmp/NzbDrone/* /opt/sonarr/bin
# chown -R sonarr:usenetserver /opt/sonarr

# vim /opt/sonarr/.config/NzbDrone/config.xml
```

```
<Config>
  <LogLevel>Info</LogLevel>
  <Port>8989</Port>
  <UrlBase>/usenetdl/sonarr</UrlBase>
  <BindAddress>*</BindAddress>
  <SslPort>9898</SslPort>
  <EnableSsl>False</EnableSsl>
  <ApiKey>*****</ApiKey>
```

```
<AuthenticationMethod>None</AuthenticationMethod>
<Branch>master</Branch>
<LaunchBrowser>True</LaunchBrowser>
<SslCertHash></SslCertHash>
<UpdateMechanism>BuiltIn</UpdateMechanism>
```

```
# /usr/bin/mono /opt/sonarr/bin/NzbDrone.exe -nobrowser -data /opt/sonarr
# vim /etc/systemd/system/sonarr.service
```

```
[Unit]
Description=Sonarr Daemon
After=syslog.target network.target

[Service]
User=sonarr
Group=usenetserver
Type=simple
ExecStart=/usr/bin/mono /opt/sonarr/bin/NzbDrone.exe -nobrowser -data
/opt/sonarr/
TimeoutStopSec=20

[Install]
WantedBy=multi-user.target
```

```
# systemctl daemon-reload # systemctl start sonarr && systemctl status sonarr # systemctl enable
sonarr </code>
```

---

## Setup Radarr- Movie Download Tool

```
cd /tmp/
wget
https://github.com/Radarr/Radarr/releases/download/v0.2.0.980/Radarr.develop
.0.2.0.980.linux.tar.gz
tar -xvf Radarr.develop.0.2.0.980.linux.tar.gz

mkdir -p /opt/radarr/bin
useradd -r radarr
usermod -a -G usenetserver radarr
cp -R Radarr/* /opt/radarr/bin/
ls -l /opt/radarr/bin/
chown -R radarr:usenetserver /opt/radarr/

vim /opt/radarr/.config/Radarr/config.xml
```

```
<Config>
  <LogLevel>Info</LogLevel>
  <Port>7878</Port>
  <UrlBase>/usenetdl/radarr</UrlBase>
  <BindAddress>*</BindAddress>
  <SslPort>9898</SslPort>
  <EnableSsl>False</EnableSsl>
  <ApiKey>*****</ApiKey>
  <AuthenticationMethod>None</AuthenticationMethod>
  <Branch>develop</Branch>
  <LaunchBrowser>True</LaunchBrowser>
  <SslCertHash></SslCertHash>
  <UpdateMechanism>BuiltIn</UpdateMechanism>
  <AnalyticsEnabled>False</AnalyticsEnabled>
```

```
vim /etc/systemd/system/radarr.service
```

```
[Unit]
Description=Radarr Daemon
After=syslog.target network.target

[Service]
User=radarr
Group=usenetserver
Type=simple
ExecStart=/usr/bin/mono /opt/radarr/bin/Radarr.exe -nobrowser -data
/opt/radarr
TimeoutStopSec=20
KillMode=process
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

```
systemctl start radarr && systemctl status radarr
systemctl enable radarr
```

---

## Setup Lidarr - Music Download Tool

```
Install / compile Cromaprint dependency:
# yum install git ffmpeg ffmpeg-devel fftw taglib boost cmake cmake3

# cd /tmp/
```



```
# git clone https://bitbucket.org/acoustid/chromaprint.git
# cd chromaprint/
# ls -la
# cmake3 -DCMAKE_BUILD_TYPE=Release -DBUILD_TOOLS=ON .
# make
# make install
# which fpcalc
# /bin/fpcalc

# cd /tmp/
# wget
https://github.com/lidarr/Lidarr/releases/download/v0.4.0.524/Lidarr.develop
.0.4.0.524.linux.tar.gz
# tar -xvf Lidarr.develop.0.4.0.524.linux.tar.gz
# ls -la Lidarr/
# mkdir -p /opt/lidarr/bin
# cp -R /tmp/Lidarr/* /opt/lidarr/bin/

# ls -la /opt/lidarr/bin/
# useradd -r lidarr
# usermod -a -G usenetserver lidarr
# chown -R lidarr:usenetserver /opt/lidarr/

# vim /opt/lidarr/.config/Lidarr/config.xml
```

```
<Config>
  <LogLevel>Info</LogLevel>
  <Port>8686</Port>
  <UrlBase>/usenetdl/lidarr</UrlBase>
  <BindAddress>*</BindAddress>
  <SslPort>6868</SslPort>
  <EnableSsl>False</EnableSsl>
  <ApiKey>*****</ApiKey>
  <AuthenticationMethod>None</AuthenticationMethod>
  <Branch>develop</Branch>
  <LaunchBrowser>True</LaunchBrowser>
  <SslCertHash></SslCertHash>
  <UpdateMechanism>BuiltIn</UpdateMechanism>
  <AnalyticsEnabled>False</AnalyticsEnabled>
```

```
# /usr/bin/mono /opt/lidarr/bin/Lidarr.exe -nobrowser -data /opt/lidarr
CTRL + C
```

```
# vim /etc/systemd/system/lidarr.service
```

```
[Unit]
```

```
Description=Lidarr Daemon
After=syslog.target network.target
```

```
[Service]
User=lidarr
Group=usenetserver
Type=simple
ExecStart=/usr/bin/mono /opt/lidarr/bin/Lidarr.exe -nobrowser -data
/opt/lidarr
TimeoutStopSec=20
KillMode=process
Restart=on-failure
```

```
[Install]
WantedBy=multi-user.target
```

```
# systemctl daemon-reload
# systemctl start lidarr.service && systemctl status lidarr.service
# systemctl enable lidarr.service
```

## Finishing Server Setup

Last update: **2019/07/10 12:41**