# Container basiertes Webhosting - Dokumentation für alte Umgebung (outdated)

Mit dem „Container-Haven“ bietet das Datenkollektiv ein Container-basiertes Webhosting an. Container sind eine Art der Virutalisierung, die sich gegenüber einer kompletten Virtualisierung durch deutlich bessere Performance auszeichnet und sehr wenig Ressourcen benötigt. Dies ermöglicht, eine Vielzahl von Containern parallel auf einem Server zu betreiben. Die einzelnen Container verhalten sich dabei wie eigenständige Server.

Gleichzeitig sorgen wir im Hintergrund für die nötigen Betriebssystem-Updates. Sie müssen sich als Nutzer_in also nur um Ihre Web-Applikationen kümmern - sofern Sie nicht auch dort z.B. das vorinstallierte Wordpress nutzen. Auch dort kümmern wir uns um die Updates.

Dies ist sozusagen ein **Mittelweg** zwischen **„einfachem Webspace“** und einem **V-Server** mit den folgenden Vorteilen:

- Die Dateien in einer Instanz, sind vollständig gegenüber anderen Instanzen isoliert
- Bei hoher Sicherheit haben die einzelnen User große individuelle Konfigurationsmöglichkeiten
- Wir kümmern uns um Sicherheitsupdates
- Ein tägliches Backup der Daten erfolgt automatisch

Wenn Sie sich für Container-basiertes Webhosting interessieren, sprechen Sie uns an.

## Zugangsdaten

Als Zugangsdaten sind notwendig:

- URL, unter der der Container verwaltet wird - üblicherweise der Domain-Name wie z.B. example.org
- ein Username für den Container (in der Regel „user“)
- ein Username für den „Container-Hafen“ (für das direkte ssh-Login, in der Regel der Domain-Name, in dem „.“ durch „_“ ersetzt ist, also z.B. example_org)
- ein Username für die Mysql-Datenbanken (in der Regel der Domain-Name, in dem „.“ durch „_“ ersetzt ist, also z.B. example_org)
- ein Passwort (anfangs identisch gesetzt: User-Passwort u. Datenbank-Passwort

Für einige Zwecke (sFTP, direkter SSH-Zugriff ist ein privater Schlüssel notwendig. Der zugehörige öffentliche Schlüssel muss auf den Servern entsprechend konfiguriert werden. Zum Einrichten bitte an den Support wenden.

## Zugang zum Container

Es gibt mehrere Möglichkeiten des Zugangs zum Container. Datei-Up- und Download lässt sich am

einfachsten über Webdav vornehmen. Vollen Zugang zum Container kann über „ssh“ erfolgen. Dateiupload über sFTP ist möglich - allerdings nur in Verbindung mit einem ssh-Tunnel.

## ssh - Secure-Shell

Der Container selbst ist nicht direkt von außen zu erreichen, sondern nur über einen weiteren Server - unseren Containerhafen. Damit das klappt, benötigen wir einen „öffentlichen SSH-Schlüssel“ von Ihnen.

Damit das Login klappt, muss „ssh Agent Forwarding“ aktiviert sein. Entweder als Option auf der Kommandozeile mit `-A` bei OpenSSH oder als Konfigurationsparameter in der ssh_config. Das kann z.B. bei OpenSSh erreicht werden, in dem entweder in die globale `/etc/ssh_config` oder die private `~/.ssh/config` Konfiguration folgendes eingefügt wird:

```
host *
    ForwardAgent yes
```

### Direkter Zugriff auf den Container

Falls entsprechend konfiguriert, kann der Container direkt mit

```
ssh -A username@container.datenkollektiv.net
```

erreicht werden.

Bitte beachten: Über diesen Weg ist nur ssh möglich. Kopieren ist weder über sFTP noch über scp oder sshfs möglich. Dies liegt an der Konfiguration auf unseren Servern, durch die der Zugang nur über einen „SSH-Proxy“ möglich ist.

### Zugriff über einen SSH-Tunnel

Um auch scp oder sFTP zu ermöglichen, muss ein SSH-Tunnel gelegt werden. Dies lässt sich auf Linux-Systemen einfach mit

```
ssh -L 2222:test-001:22 username@container.datenkollektiv.net
```

Der Server ist dann unter `user@localhost` auf Port 2222 zu erreichen. Also z.B.:

```
scp -P 2222 file user@localhost:
```

### Webdav

Ausgewählte Verzeichnisse des Containers sind über Webdav zu erreichen. Die Webdav-Adresse lautet dabei in der Regel:

```
https://webdav.example.org/
```

Die Zugangsdaten lauten:

- Loginname ist immer: `user`
- Passwort ist das nach dem Anlegen übermittelte

Der Zugriff erfolgt dabei auf das Verzeichnis `/srv` auf dem Container.

Achtung: Bei Webdav kann es Probleme mit Leerzeichen und Sonderzeichen in Dateinamen geben. Diese gilt es in jedem Fall zu vermeiden. Insbesondere Dateien mit Leerzeichen können nicht angelegt werden. Leider ist das bei einigen Dateimanagern die Standard-Einstellung („Neue Datei“). In diesem Fall hilft, eine Datei lokal anzulegen und zu kopieren.

Andere Verzeichnisse können über Symlinks erreicht werden. Per SSH-Zugriff können auf diese Weise beliebige Verzeichnisse per Webdav verfügbar gemacht werden.

## Dateien hochladen

Am einfachsten funktioniert das per Webdav. Für alle Plattformen gibt es dazu Webdav-kompatible Dateibrowser. Unter Linux-Desktops funktionert das z.B. mit der Funktion „Mit Server verbinden“.

Auf diese Weise können Dateien kopiert oder bearbeitete werden als ob sie auf dem lokalen Rechner liegen würden. Neben der verbesserten Sicherheit gegenüber „ftp“ ist dies auch intuitiver und einfacher.

Für Mac-Os Nutzer: Uns wurde berichtet, dass integrierte Webdav Support des „Finders“ verschiedene Probleme macht. Mit anderen generischen Webdav-Clients funktioniert es aber auch unter Mac-Os.

## Datenbanken konfigurieren

Die Datenbanken sind unter folgender URL zu erreichen:

```
https://mysql.datenkollektiv.net/dknmysqladmin/
```

Login mit Datenbank-Username und Passwort. Dort können beliebige Datenbanken erstellt werden, deren Namen mit dem Usernamen gefolgt von einem „_“ beginnen müssen. Also z.B.

```
example_org_wordpress
```

## Wordpress installieren

In den Containern ist eine Wordpress-Instanz vorinstalliert. Sollte diese noch nicht initialisiert sein, so kann das über einen SSH-Zugang erfolgen. Im Home-Verzeichnis findet sich eine README Datei, in denen die Befehle stehen, die ausgeführt werden müssen, um Datenbank und Wordpress Instanz zu konfigurieren.

## Eigene html-Seiten / Php-Skripte

Der Webroot liegt unter `/var/www/html/` - und ist per webdav zugänglich. Alle Dateien und Skripte können dort hochgeladen werden.

## E-Mail-versand ermöglichen

Wir erlauben per Default keinen unauthentifizierten E-Mail-Versand aus den Containern mit dem normalen Unix-Mail-Kommando. Erstens würde das bedeuten, dass falsch Konfigurierte Server schnell zu Spam-Schleudern würden und außerdem könnten damit E-Mails mit falschen Absendern etc. verschickt werden. Stattdessen kann ein E-Mail Account konfiguriert werden, über den alle E-Mails aus den Containern verschickt werden. Das ist auch die zuverlässigste Methode, bei der es seltener dazu kommt, dass andere Mailserver die Mails als Spam abweisen, weil der Hostname oder der E-Mail-Absender nicht bekannt ist.

Dafür ist innerhalb der Container das Programm ssmtp installiert. Dieses muss noch konfiguriert werden, das lässt sich ebenfalls über webdav vornehmen, die Konfigurationsdatei sind unter /etc/ssmtp/* verlinkt.

Zur Konfiguration von ssmtp siehe auch z.B.: https://wiki.archlinux.org/index.php/SSMTP

**Beispielkonfiguration:**

Account an den alle System E-Mails gehen sollen: your_receiving_account@notraces.net Account über den die E-Mails verschickt werden: youroutgoingaccount@notraces.net (username/password als AuthUser/AuthPass eintragen) Mailserver, auf dem der Account existiert, über den die E-Mails verschickt werden mit Port: mail.datenkollektiv.net:587

Per Webdav die Datei `/etc/ssmtp/ssmtp.conf` öffnen und editieren:

```
#
# Config file for sSMTP sendmail
```

```
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=your_receiving_account@notraces.net

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=mail.datenkollektiv.net:587

# Where will the mail seem to come from?
# rewriteDomain=example.org

# The full hostname (the container-default-hostname)
hostname=your.container.hostname

# Use SSL/TLS before starting negotiation
UseTLS=Yes
UseSTARTTLS=Yes

# Username/Password # das muss mit dem konfigurierten Account in revaliases
korrespondieren.
AuthUser=USERNAME
AuthPass=VERY-SECRET_PASSWORD

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=NO
```

Jetzt muss noch die Datei `/etc/ssmtp/revaliases` konfiguriert werden. In ihr wird jedem lokalen Nutzer ein Mailaccount zugeordnet. Wichtig ist dabei neben root auch der Nutzer www-data - da über ihn normalerweise Mails vom Webserver verschickt werden.

```
# sSMTP aliases
#
# Format:       local_account:outgoing_address:mailhub
#
# Example: root:your_login@your.domain:mailhub.your.domain[:port]
# where [:port] is an optional port number that defaults to 25.

root:youroutgoingaccount@notraces.net
www-data:youroutgoingaccount@notraces.net
```

# Webserver konfigurieren

Fortgeschrittene User_innen können über einen SSH-Zugriff auch die Webserver-Konfiguration selbst verändern. Dazu sind Kenntnisse in der Konfiguration des Nginx-Webservers nötig. Außerdem muss beachtet werden, dass der Webserver hinter einem Webproxy liegt - und z.B. nicht auf alle Header-Variablen direkt zugegriffen werden kann. Außerdem geschieht die Verbindung zum Webproxy

grundsätzlich über Port 80 und http. Sämtliche Konfiguration von Zertifikaten geschieht auf dem Proxy-Server durch die Administratoren des datenkollektiv.

Der Default-User hat die Rechte, die Dateien unter /etc/nginx/sites-available und /etc/nginx/sites-enabled/ zu bearbeiten und per

```
sudo service nginx restart
```

oder

```
sudo service nginx reload
```

den Webserver neu zu starten.