

Einrichtung Nextcloud unter Debian 10

Dieser Artikel ist aktuell in Bearbeitung und kann so noch nicht verwendet werden!!!!!!

Ausgangssituation

Auf der Suche nach der Möglichkeit private Bilder etc. nicht in einer öffentlichen Cloud speichern zu müssen landet man früher oder später bei der Variante eine eigene Cloud aufzubauen. Nach einigen Experimenten von owncloud über nextcloud auf einem QNAP-System habe ich beschlossen das Ganze sauber auf Basis der aktuellen nextcloud-Version aufzubauen. Die Grundlage bietet ein Microserver von Wortmann mit einem RAID5-Verbund von 4 Festplatten. Unter der darauf laufenden vmWare ESXi werden wir jetzt nextcloud unter der aktuellen Debian 10.8 aufbauen.

Für die nachstehenden Aktionen benötigen Sie root-Zugriff auf Ihrem Server

So geht's

Grundinstallation DEBIAN unter vmware ESXi

Zunächst installieren wir in einer virtuellen Maschine unter vmware ESXi eine Standardinstallation von DEBIAN (aktuell debian-10.8.0-amd64-netinst.iso). Nachdem während der Installation keine Partitionierung vorgegeben wurde, in der /var die größte Partition ist, nehmen wir am Besten die Variante 'Alle Dateien in einer Partition (Empfohlen für Anfänger)'. Damit haben wir keine Einschränkungen, die uns zu einem späteren Zeitpunkt eine Partition zu klein werden lassen.

Bei der Softwareauswahl haben wir das debian desktop environment deaktiviert, web und SSH server sowie Standard-Systemwerkzeuge aktiviert. Den restlichen Komponenten installieren wir wie nachstehen beschrieben.

System auf aktuellen Stand bringen

Melden Sie sich zunächst mit dem Root-Account am Server an und bringen Sie Ihren Server auf den aktuellen Stand einer stabilen Version von DEBIAN. Nutzen Sie hierzu folgende Commands:

```
apt-get update -yENTER  
apt-get upgrade -yENTER
```

Nach dem erfolgreichen Update Ihres Servers starten Sie diesen neu um das Update abzuschließen:

```
rebootENTER
```

Nach dem Neustart melden Sie sich mit dem root-Account an und fahren Sie mit dem nächsten Schritt fort.

IP-Adresse festlegen

Nachdem wir bei Servern nur ungern IP-Adressen über DHCP zuweisen (über eine feste MAC-Zuordnung wäre diese natürlich auch möglich) ändern wir nun die IP-Adresse des Netzadapters.

Über `ip a` bekommen wir die aktuelle Konfiguration angezeigt:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens192: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP
group default qlen 1000
    link/ether 00:0c:29:24:c8:39 brd ff:ff:ff:ff:ff:ff
    inet xx.xx.xx.xx/25 brd xx.xx.xx.xx scope global dynamic ens192
        valid_lft 85573sec preferred_lft 85573sec
    inet6 fe80::20c:29ff:fe24:c839/64 scope link
        valid_lft forever preferred_lft forever
```

Über `vi /etc/network/interfaces` können wir nun die IP-Adresse festlegen indem wir den Eintrag entsprechend festlegen:

```
# The primary network interface
allow-hotplug ens192
iface ens192 inet static
    address x.x.x.x
    netmask 255.255.255.0
    broadcast x.x.x.x
    gateway x.x.x.x
```

Jetzt speichern wir das Ganze und führen einen erneuten Reboot aus.

SSH-Zugang einrichten

Um den root-Account nicht für SSH freizugeben, jetzt noch `sudo` installieren (`apt-get install sudo`) und den bei der DEBIAN-Installation angegebenen Hauptbenutzer oder einen beliebigen weiteren Benutzer entsprechend berechtigen.

Und jetzt geht's los mit der eigentlichen Installation von Nextcloud.

LAMP-Server installieren

Nextcloud benötigt Apache, MySQL und PHP. Diese müssen wir nun zunächst auf dem Server installieren.

Im ersten Schritt installieren wir Apache, MariaDB und Sudo über nachstehendes Command:

```
apt-get install apache2 mariadb-server apt-transport-https -y ENTER
```

Nach erfolgreicher Installation der Pakete starten Sie Apache und MariaDB und aktivieren Sie deren Start beim Booten des Systems:

```
systemctl start apache2 ENTER  
systemctl enable apache2 ENTER  
systemctl start mariadb ENTER  
systemctl enable mariadb ENTER
```

Als nächsten Schritt installieren wir PHP in der aktuellsten Version 7.3 die von Nextcloud unterstützt wird inkl. aller benötigten Pakete:

```
apt install php7.3 libapache2-mod-php7.3 php7.3-xml php7.3-curl php7.3-gd  
php7.3-cgi php7.3-cli php7.3-zip php7.3-mysql php7.3-mbstring php7.3-intl  
php7.3-imagick -y ENTER
```

Nach erfolgreicher Installation editieren wir PHP.INI (natürlich mit vorhergehender Anfertigung einer Sicherungskopie):

```
vi /etc/php/7.3/apache2/php.ini ENTER
```

und setzen hier die erforderlichen/sinnvollen Werte:

```
date.timezone = Europe/Berlin  
max_execution_time = 3600  
memory_limit = 512M  
post_max_size = 4G  
upload_max_filesize = 4G
```

Die 4G haben sich als praxistauglich erwiesen, wenn auch Videos hochgeladen werden soll. Ansonsten sollten auch Werte um 512M ausreichend sein 😊

Damit haben wir die Grundinstallation der LAMP-Servers abgeschlossen und können nun mit der Konfiguration/Einrichtung der Datenbank fortfahren.

Abschließend testen wir ob die PHP-Installation erfolgreich war. Hierzu legen wir eine kleine Datei info.php im Verzeichnis /var/www/html/info.php über das Kommando

```
vi /var/www/html/info.php ENTER
```

```
<?php phpinfo(); ?>
```

Nachdem wir die Datei gespeichert haben, öffnen wir einen Browser und geben die URL

```
http://beispielserver.de/info.php ENTER
```

ein.

Es sollte dann die PHP-Informationen des Servers zu sehen sein.

Konfiguration MariaDB

Nach der erfolgreichen Installation von MariaDB wird empfohlen das nachstehende Sicherheitsscript auszuführen, welches einige unsichere Defauteinstellungen entfernt und den Zugriff auf das Datenbanksystem untersagt.

```
mysql_secure_installation
```

The above security script will take you through a series of following questions where you can make some changes to your MariaDB setup as shown.

```
# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user.  If you've just installed MariaDB, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
... Success!
```

```
By default, MariaDB comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] y
```

```
- Dropping test database...
```

```
... Success!
```

```
- Removing privileges on test database...
```

```
... Success!
```

```
Reloading the privilege tables will ensure that all changes made so far will take effect immediately.
```

```
Reload privilege tables now? [Y/n] y
```

```
... Success!
```

```
Cleaning up...
```

```
All done! If you've completed all of the above steps, your MariaDB installation should now be secure.
```

```
Thanks for using MariaDB!
```

Nachdem MariaDB gesichert ist, loggen wir uns nun in die Konsole ein:

```
mysql -u root -p ENTER
```

Geben Sie das root-Passwort ein wenn Sie danach gefragt werden und legen Sie dann die nextcloud-Datenbank an

```
CREATE DATABASE nextclouddb; ENTER
```

Anschließend erstellen Sie einen Benutzer für Nextcloud über folgenden Befehl

```
CREATE USER 'nextcloud'@'localhost' IDENTIFIED BY '<password>'; ENTER
```

wobei Sie <password> durch das von Ihnen gewünschte, sichere Kennwort ersetzen.

Jetzt setzen wir die Rechte für den neuen Benutzer auf alle Objekte für die Datenbank nextclouddb

```
GRANT ALL PRIVILEGES ON nextclouddb.* to 'nextcloud'@'localhost'; ENTER
```

und schreiben das Ganze persistent über

```
FLUSH PRIVILEGES; ENTER
```

Nachdem wir nun alle vorbereitenden Kommandos ausgeführt haben, verlassen wir MariaDB durch

```
quit ENTER
```

und fahren nun nach der erfolgreichen Anlage der Datenbank mit der Installation von Nextcloud fort.

Installation von Nextcloud

Für die Installation selbst, werden wir den Nextcloud Web-Installer verwenden. Zunächst aber werden wir das notwendige Umfeld hierfür installieren.

Zuerst das Verzeichnis für die Installation von Nextcloud

```
mkdir /var/www/nextcloudENTER  
chown www-data:www-data /var/www/nextcloudENTER  
chmod 750 /var/www/nextcloudENTER
```

Jetzt noch das Datenverzeichnis, in dem Nextcloud dann die hochgeladenen Daten ablegen wird

```
mkdir -p /var/nextcloud/dataENTER  
chown www-data:www-data /var/nextcloud/dataENTER  
chmod 750 /var/nextcloud/dataENTER
```

Abschließend erstellen wir noch eine Apache-VirtualHost-Datei für Nextcloud. Hierzu erzeugen wir die Datei nextcloud.conf im entsprechenden Verzeichnis

```
vi /etc/apache2/sites-available/nextcloud.confENTER
```

und fügen folgende Zeilen ein

```
<VirtualHost *:80>  
ServerAdmin admin@<beispiel.de>  
DocumentRoot "/var/www/nextcloud"  
ServerName <Beispiel.de>  
<Directory "/var/www/nextcloud/">  
Options MultiViews FollowSymlinks  
  
AllowOverride All  
Order allow,deny  
Allow from all  
</Directory>  
TransferLog /var/log/apache2/nextcloud_access.log  
ErrorLog /var/log/apache2/nextcloud_error.log  
</VirtualHost>
```

Ersetzen Sie dabei **<Beispiel.de>** durch Ihren Domainnamen. Nachdem Sie die Datei gesichert haben deaktivieren wir den Standard Virtualhost und aktivieren Nextcloud.

```
a2dissite 000-defaultEnter  
a2ensite nextcloudEnter
```

und schliessen das Ganze mit `systemctl reload apache2``Enter` ab.

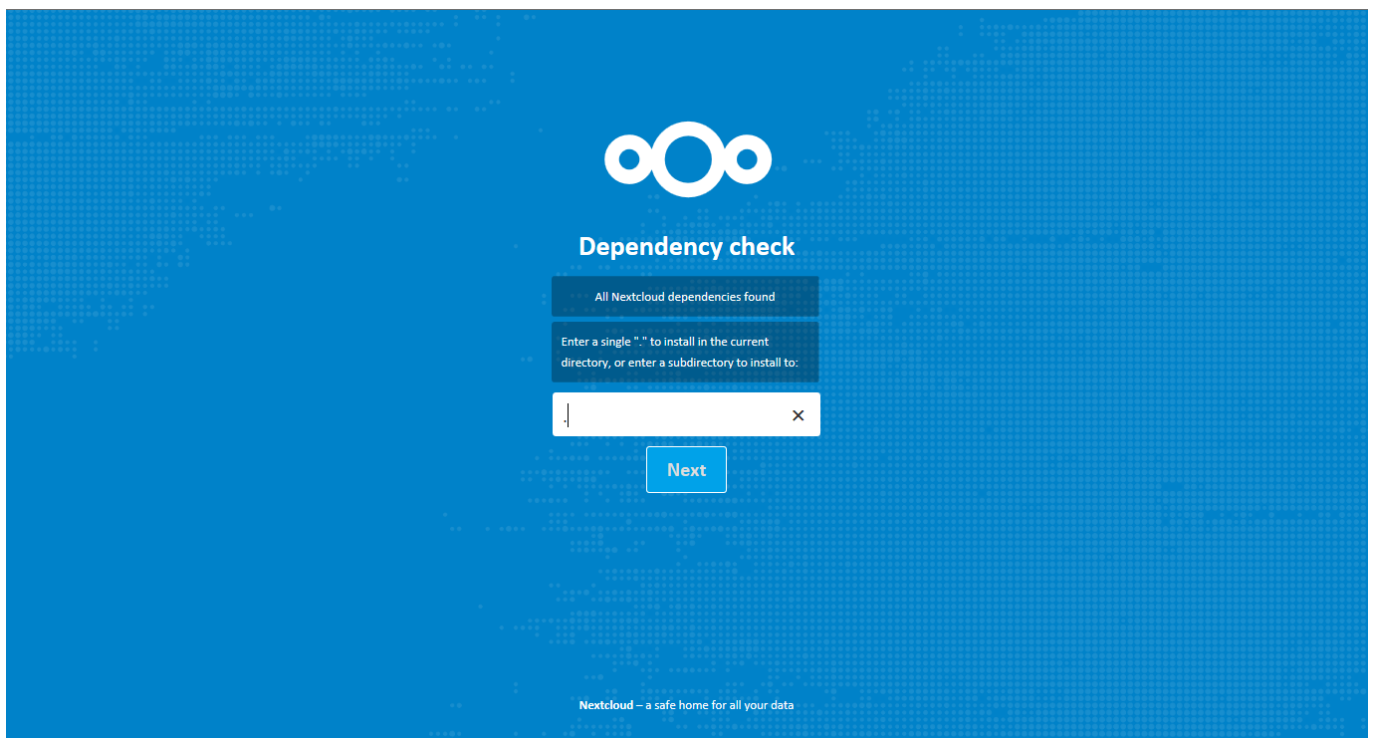
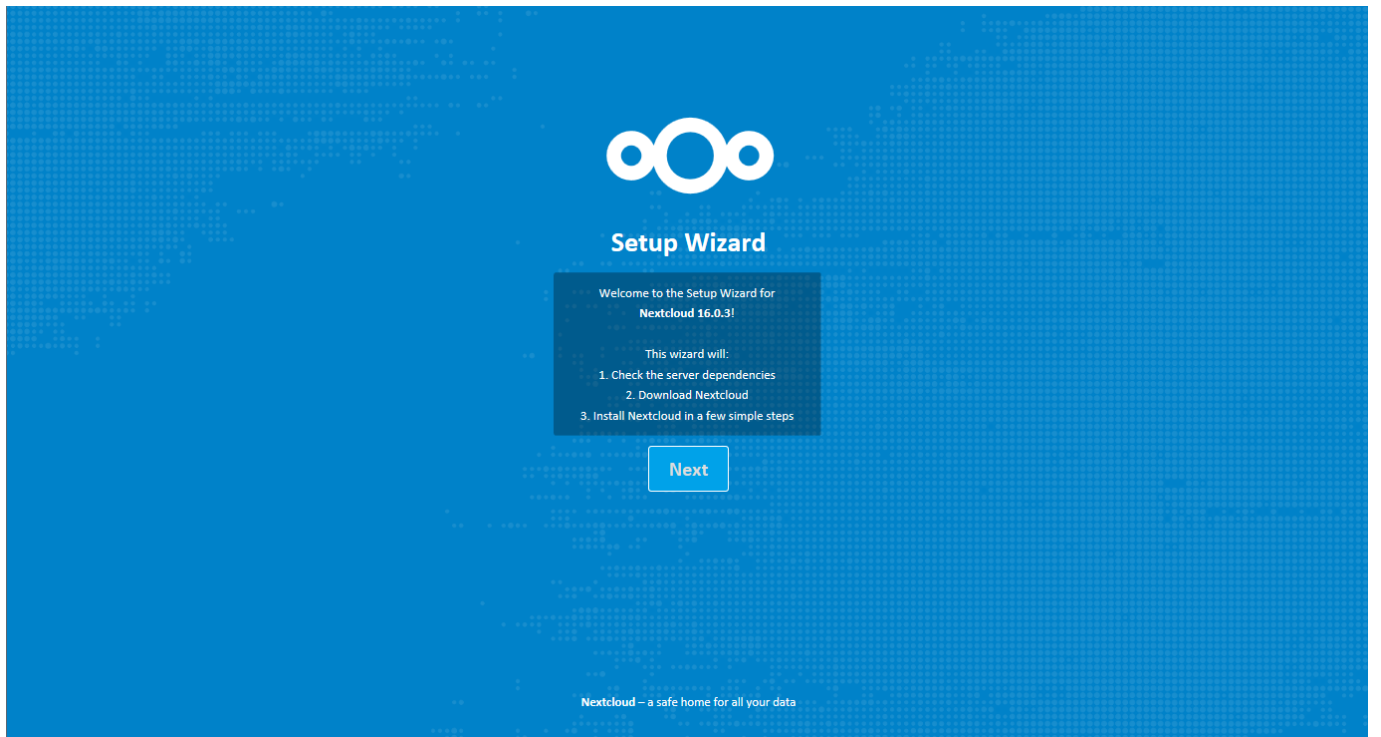
*** ===== SSL über Let's encrypt aktivieren ===== Um SSL über Let's encrypt zu aktivieren installieren wir zunächst den den Let's encrypt cerbot Client apt-get install certbot python3-certbot-apache -y`ENTER` und aktivieren das Apache SSL Modul über folgenden Befehl a2enmod ssl`ENTER` Abschließend starten wir den Apache-Service neu systemctl restart apache2`ENTER` Jetzt können wir ein freies SSL-Zertifikat von let's encrypt**

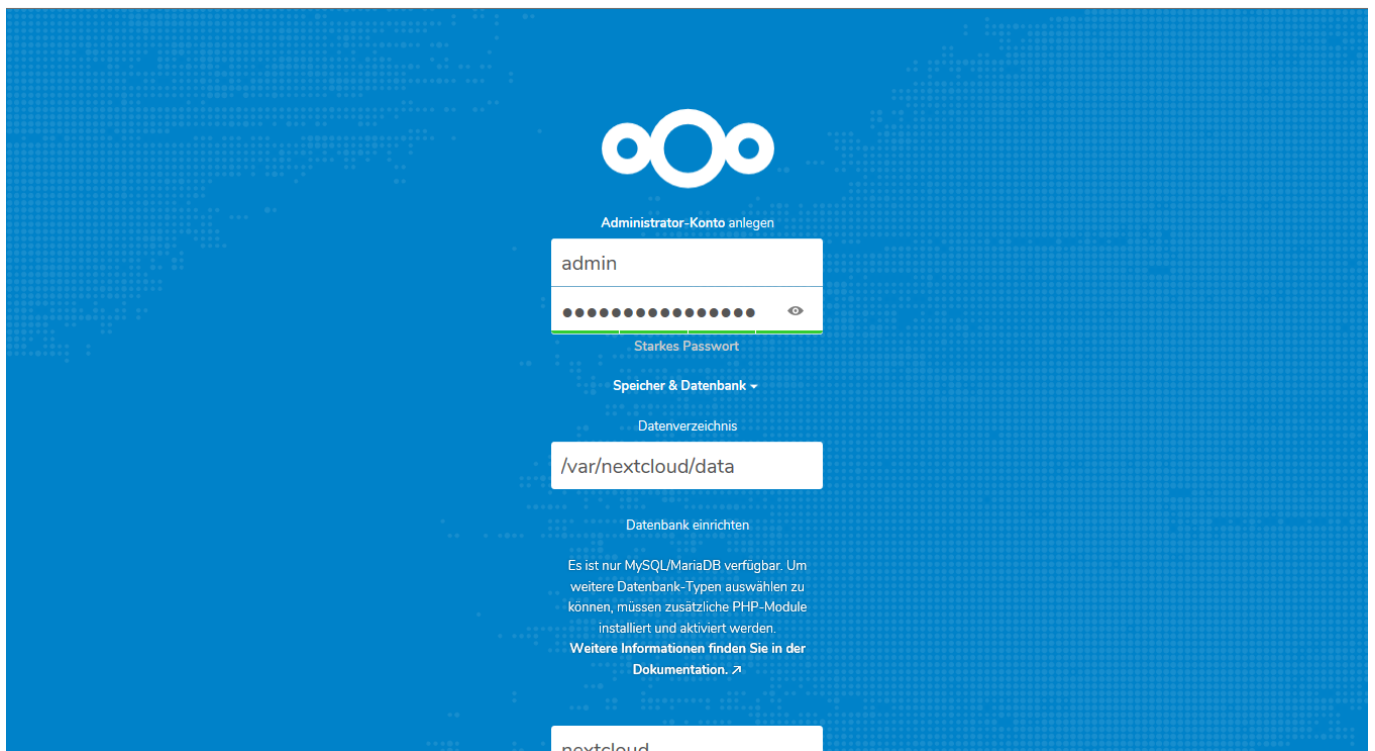
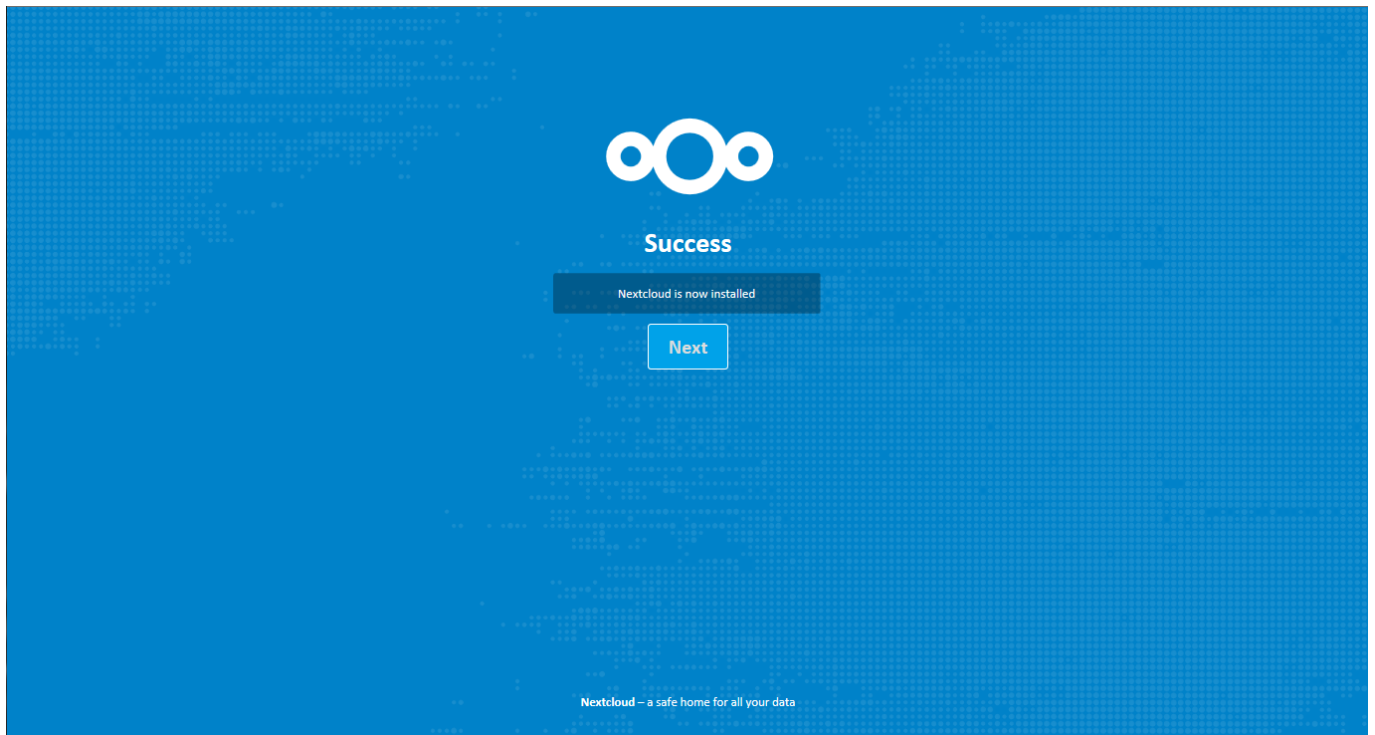
anfordern und über certbot den Apache VHOST konfigurieren. Bitte dabei beachten, dass Ihre Domain bereits vom Internet aus erreichbar und bereits ein DNS-A-Record darauf zeigen muss um let's encrypt nutzen zu können. Nachstehendes Kommando fordert nun ein neues SSL-Zertifikat an: certbot -d beispiel.de -apache -agree-tos -m info@beispiel.de **ENTER** wobei hier beispiel.de durch den entsprechenden Domainnamen und info@beispiel.de durch die entsprechende E-Mail-Adresse zu ersetzen ist. <code>

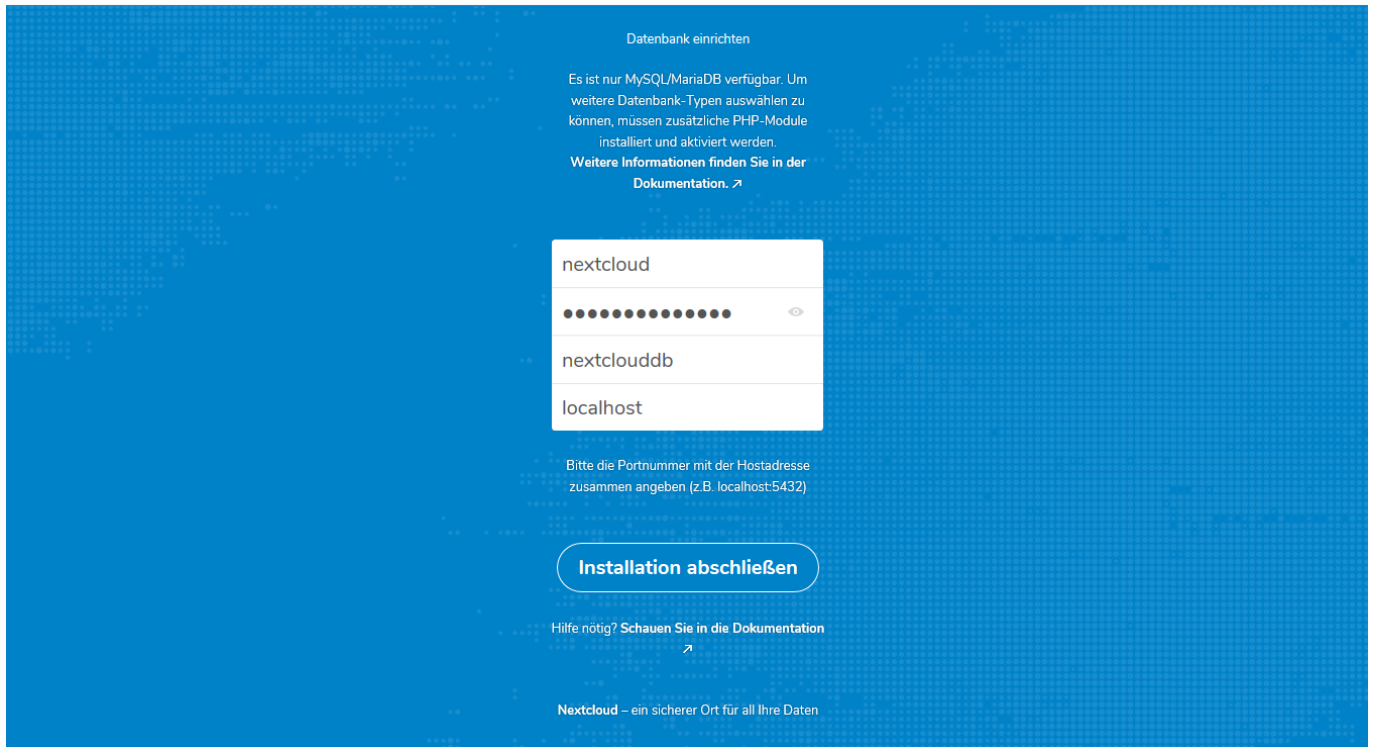
```

root@storage:/# certbot -d shbcloud.s4dip.net -apache -agree-tos -m
theo.schoeberl@germanmx.de Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator apache, Installer apache Obtaining a new certificate
Performing the following challenges: http-01 challenge for shbcloud.s4dip.net Enabled
Apache rewrite module Waiting for verification... Cleaning up challenges Created an SSL
vhost at /etc/apache2/sites-available/nextcloud-le-ssl.conf Deploying Certificate to
VirtualHost /etc/apache2/sites-available/nextcloud-le-ssl.conf Enabling available site:
/etc/apache2/sites-available/nextcloud-le-ssl.conf Please choose whether or not to redirect
HTTP traffic to HTTPS, removing HTTP access. -----
----- 1: No redirect - Make no further changes to the webserver configuration. 2:
Redirect - Make all requests redirect to secure HTTPS access. Choose this for new sites, or
if you're confident your site works on HTTPS. You can undo this change by editing your
web server's configuration. ----- Select
the appropriate number [1-2] then [enter] (press 'c' to cancel): 2 Enabled Apache rewrite
module Redirecting vhost in /etc/apache2/sites-enabled/nextcloud.conf to ssl vhost in
/etc/apache2/sites-available/nextcloud-le-ssl.conf -----
----- Congratulations! You have successfully enabled https://shbcloud.s4dip.net
You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=shbcloud.s4dip.net -----
----- IMPORTANT NOTES: - Congratulations! Your certificate and
chain have been saved at: /etc/letsencrypt/live/shbcloud.s4dip.net/fullchain.pem Your key
file has been saved at: /etc/letsencrypt/live/shbcloud.s4dip.net/privkey.pem Your cert will
expire on 2021-06-13. To obtain a new or tweaked version of this certificate in the future,
simply run certbot again with the "certonly" option. To non-interactively renew *all* of
your certificates, run "certbot renew" - If you like Certbot, please consider supporting our
work by: Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate Donating to EFF:
https://eff.org/donate-le </code>

```







Schlagwörter

nextcloud qnap Installation lamp

From:

<https://wiki.tssystem.de/> - TS Systems - DokuWiki

Permanent link:

https://wiki.tssystem.de/doku.php?id=cloud:nextcloud_on_debian&rev=1615838162

Last update: **15.03.2021 20:56**

