

# Laborbericht - NVS - 5CHIF

Name: Juri Schreib

Datum: 2017-03-21

Ziel: Erfüllung der Aufgabenstellung

## Virtuelle Maschinen erstellen

Für das Test-Set-up werden zunächst 2 Lxc Container auf meinem privaten Proxmox Hypervisor erstellt. Einige Services, wie ssh und VNC werden in der Initialen Konfiguration von Proxmox automatisch aktiviert.

### Acadia

The screenshot shows the 'Create: LXC Container' dialog box in Proxmox, with the 'General' tab selected. The dialog has a title bar with a close button (X) and a 'Help' button. Below the title bar are tabs for 'General', 'Template', 'Root Disk', 'CPU', 'Memory', 'Network', 'DNS', and 'Confirm'. The 'General' tab contains the following fields:

- Node:** oak (dropdown menu)
- VM ID:** 100 (spin box)
- Hostname:** kmu-Acadia (text input)
- Unprivileged container:**
- Resource Pool:** (empty dropdown menu)
- Password:** (password input field with dots)
- Confirm password:** (password input field with dots)
- SSH public key:** ssh-rsa AAAAB3NzaC1yc2E/ (text input)

There is a blue button labeled 'Load SSH Key File' below the SSH public key field. At the bottom of the dialog are 'Back' and 'Next' buttons.

### Create: LXC Container

General **Template** Root Disk CPU Memory Network DNS Confirm

Storage:

Template:

### Create: LXC Container

General Template **Root Disk** CPU Memory Network DNS Confirm

Storage:  ACLs:

Disk size (GB):  Enable quota:

## Create: LXC Container



General   Template   Root Disk   **CPU**   Memory   Network   DNS   Confirm

Cores:

Help

Back

Next

## Create: LXC Container



General   Template   Root Disk   CPU   **Memory**   Network   DNS   Confirm

Memory (MB):

Swap (MB):

Help

Back

Next

Create: LXC Container

General Template Root Disk CPU Memory **Network** DNS Confirm

Name (i.e. eth0):  IPv4:  Static  DHCP

MAC address:  IPv4/CIDR:

Bridge:  Gateway (IPv4):

VLAN Tag:  IPv6:  Static  DHCP  SLAAC

Rate limit (MB/s):  IPv6/CIDR:

Firewall:  Gateway (IPv6):

[Help](#) [Back](#) [Next](#)

Create: LXC Container

General Template Root Disk CPU Memory Network **DNS** Confirm

DNS domain:

DNS server 1:

DNS server 2:

DNS server 3:

[Back](#) [Next](#)

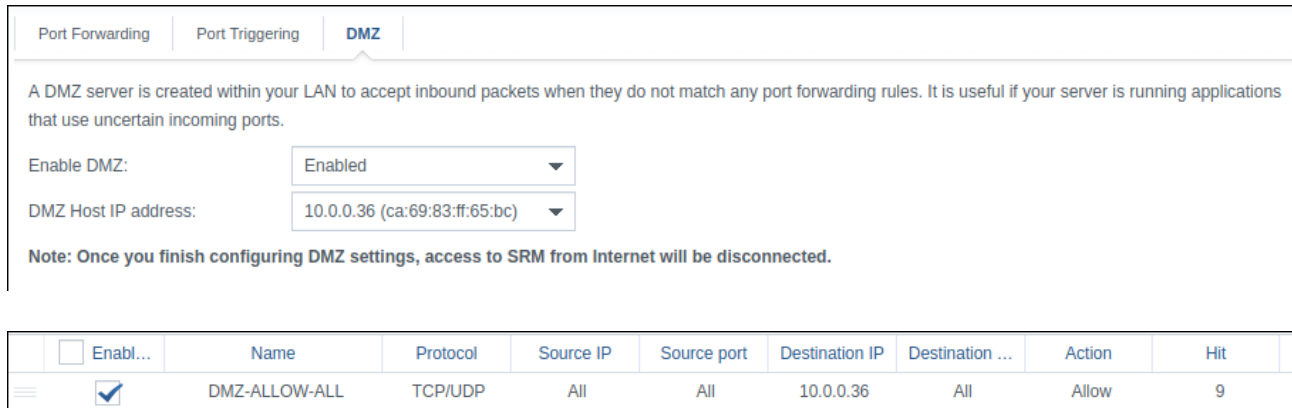
## Badlands

Es wird dieselbe Konfiguration ausgeführt, mit dem Unterschied, dass auf Badlands die

IP-Adresse 10.0.0.37 und die Bezeichnung kmu-badlands gewählt wird.

# Internetverbindung für Acadia öffentlich stellen

Im Nat Device wird ein neues DMZ-Gerät und ein Firewall-Eintrag hinzugefügt:



<input type="checkbox"/>	Enabl...	Name	Protocol	Source IP	Source port	Destination IP	Destination ...	Action	Hit
<input checked="" type="checkbox"/>		DMZ-ALLOW-ALL	TCP/UDP	All	All	10.0.0.36	All	Allow	9

Danach ist der V-Server Acadia öffentlich erreichbar:

```
bujuhu ~ ssh root@versandkostenfrei.kaufensie.jetzt
Linux kmu-Acadia 4.4.35-1-pve #1 SMP Fri Dec 9 11:09:55 CET 2016 x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Mar 21 05:59:22 2017 from 80-109-104-102.cable.dynamic.surfer.at
root@kmu-Acadia:~#
```

## Webanwendungen auf Acadia installieren und verfügbar machen.

OpenSSH ist bereits vorinstalliert, daher muss das nicht mehr manuell aufgesetzt werden.

Vor der Installation wird das System mit `apt-get update; apt-get upgrade` auf den aktuellsten Stand gebracht

### Installation von Docker und Docker-Compose

Docker wird [nach der Anleitung der Docker Website](#) installiert.

Docker-Compose wird ebenfalls [nach der Anleitung](#) installiert.

Um zu testen ob die Anwendungen installiert sind, wird zum `testendocker -v` und `docker-compose -v` verwendet.

```
root@kmu-Acadia:~# chmod +x /usr/local/bin/docker-compose
root@kmu-Acadia:~# docker -v
Docker version 17.03.0-ce, build 3a232c8
root@kmu-Acadia:~# docker-compose -v
docker-compose version 1.11.2, build dfed245
root@kmu-Acadia:~#
```

Nach der Vollständigen Installation von Docker-Compose kann nun mit der Installation der einzelnen Komponenten begonnen werden

## Installation von Webmin

Webmin wird <http://www.debianadmin.com/install-webmin-on-debian-7-6-wheezy.html> unter Debian installiert.

Die Verschlüsselung des Webmin Miniserv wird deaktiviert, da die Verschlüsselung von Nginx übernommen wird.

nano /etc/webmin/miniserv.conf Der Parameter `ssl=1` wird auf `ssl=0` gesetzt. Danach wird Webmin neu gestartet.

```
-bash: systemctl: command not found
root@kmu-Acadia:~# service webmin status
Webmin (pid 17756) is running
root@kmu-Acadia:~#
```

## Installation von NGINX

Webmin wird ebenfalls direkt auf dem Host installiert

```
apt-get install -y nginx
```

```
root@kmu-Acadia:~# service nginx status
[ ok ] nginx is running.
```

## Welcome to nginx on Debian!

If you see this page, the nginx web server is successfully installed and working on Debian. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org)

Please use the `reportbug` tool to report bugs in the nginx package with Debian. However, check [existing bug reports](#) before reporting a new bug.

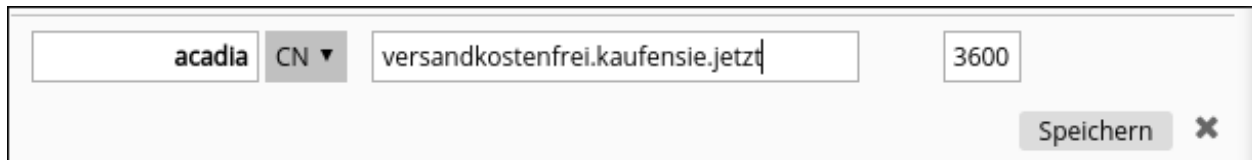
*Thank you for using debian and nginx.*

Die Installation von NGINX war erfolgreich.

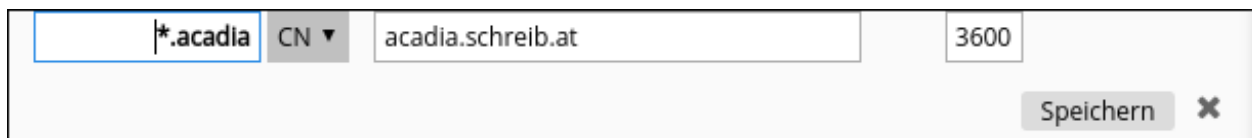
## Subdomäne erstellen

Dafür wird als erstes eine eigene Subdomain für den Server erstellt

Dazu wird beim DNS Server ein neuer CNAME Record erstellt



Der Einfachheit halber wird die Wildcard Domain `*acadia.schreib.at` erstellt, um auf alle Dienste des Servers zugreifen zu können.



## NextCloud (Docker)

NextCloud wird in einem Docker Container deployed. Dazu wird die von NextCloud [vorkonfigurierte Docker-Compose Projekt](#) genutzt.

Um das nutzen zu können wird als erstes git installiert:`apt-get install -y git`

Des weiteren wird mit `docker network create lb_web` ein internes Netzwerk erstellt

Danach wird das Projekt mit den folgenden Befehlen installiert und gestartet:

```
git clone https://github.com/indiehosters/nextcloud.git
cd nextcloud
MYSQL_ROOT_PASSWORD=ciscoclass docker-compose up
```

```
ca6ffbbcdc10: Download complete
ba8ff064032b: Download complete
ERROR: failed to register layer: ApplyLayer exit status 1 stdout: stderr: permission denied
root@kmu-Acadia:~/nextcloud#
```

Es tritt ein Fehler auf. Nach Recherche scheint das Problem zu sein, dass in Proxmox Lxc Containern kein Docker unterstützt wird. Deshalb wird NextCloud sowie gitlab nativ auf dem Server installiert.

## NextCloud (Nativ)

Als erstes wird NextCloud heruntergeladen und in das richtige Verzeichnis verschoben:

```
cd ~
```

```
wget https://download.nextcloud.com/server/releases/nextcloud-11.0.2.zip
unzip nextcloud-11.0.2.zip
mkdir /var/www/
mv nextcloud /var/www/
rm nextcloud-10.0.2.zip
chown -R www-data: /var/www/nextcloud
```

Um Nginx verwenden zu können wird auch noch ein MySQL kompatibler Server benötigt. Dafür wird MariaDB gewählt und installiert

```
sudo apt-get install -y mariadb-server
```

Das Administratorpassword der Datenbank wird auf *ciscoclass* gesetzt.

## GitLab

Gitlab wird [nach der Anleitung auf der GitLab Seite](#) installiert.

Der externe Port wird von 80 auf 8080 verändert, indem die `external_url` konfigurationsparameter in `/etc/gitlab/gitlab.rb` auf `external_url http://127.0.0.1:8080/` gesetzt wird. Danach wird gitlab mit dem Befehl `gitlab-ctl reconfigure` neugestartet.

## NGINX konfigurieren

Damit NextCloud richtig funktioniert muss erstmal php für NGINX installiert werden. Dafür werden die Pakete `php5`, `php5-cgi`, `php5-gd`, `php5-curl`, `php5-mysql` und `php5-fpm` benötigt

NGINX wird dazu genutzt, auf die einzelnen Webanwendungen mithilfe von subdomänen zugreifen zu können (sprich `webmin.acadia.schreib.at` für webmin, `git.acaida.schreib.at` für gitlab und `acadia.schreib.at` für NextCloud)

Nginx Konfigurationsdatei:

```
# Webmin
server {
    server_name webmin.acadia.schreib.at;
    listen 80;
    location / {
        proxy_redirect http://127.0.0.1:10000/ http://webmin.acadia.schreib.at/;
        proxy_pass http://127.0.0.1:10000/;
        proxy_set_header    Host    $host;
    }
}

# GitLab
server {
    server_name git.acadia.schreib.at;
    listen 80;
    location / {
        proxy_redirect http://127.0.0.1:8080/ http://git.acadia.schreib.at/;
        proxy_pass http://127.0.0.1:8080/;
        proxy_set_header    Host    $host;
    }
}
```



```

# NextCloud
server {
    listen 80;
    server_name cloud.acadia.schreib.at;

    #ssl_certificate /etc/ssl/nginx/cloud.example.com.crt;
    #ssl_certificate_key /etc/ssl/nginx/cloud.example.com.key;

    root /var/www/;

    # set max upload size
    client_max_body_size 10G;

    # Disable gzip to avoid the removal of the ETag header
    gzip off;

    # Uncomment if your server is build with the ngx_pagespeed module
    # This module is currently not supported.
    #pagespeed off;

    index index.html index.php;
    error_page 403 /core/templates/403.php;
    error_page 404 /core/templates/404.php;

    rewrite ^/.well-known/carddav /remote.php/dav/ permanent;
    rewrite ^/.well-known/caldav /remote.php/dav/ permanent;

    # The following 2 rules are only needed for the user_webfinger app.
    # Uncomment it if you're planning to use this app.
    #rewrite ^/.well-known/host-meta /public.php?service=host-meta last;
    #rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-json last;

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    location ~ ^/(build|tests|config|lib|3rdparty|templates|data)/ {
        deny all;
    }

    location ~ ^/(?!\.|autotest|occ|issue|indie|db_|console) {
        deny all;
    }

    location / {

        rewrite ^/remote/(.*) /remote.php last;

        rewrite ^(/core/doc/[^/]+/)$ $1/index.html;

        try_files $uri $uri/ =404;
    }

    location ~ \.php(?:$|/) {

```

```

fastcgi_param HTTP_PROXY "";

fastcgi_pass unix:/var/run/php5-fpm.sock;
fastcgi_index index.php;
include fastcgi_params;
}

# Adding the cache control header for js and css files
# Make sure it is BELOW the location ~ \.php(?:$|/) { block
location ~* \.(?:css|js)$ {
add_header Cache-Control "public, max-age=7200";
# Add headers to serve security related headers
add_header Strict-Transport-Security "max-age=15768000; includeSubDomains;
preload;";
add_header X-Content-Type-Options nosniff;
add_header X-Frame-Options "SAMEORIGIN";
add_header X-XSS-Protection "1; mode=block";
add_header X-Robots-Tag none;
add_header X-Download-Options noopen;
add_header X-Permitted-Cross-Domain-Policies none;
# Optional: Don't log access to assets
access_log off;
}

# Optional: Don't log access to other assets
location ~* \.(?:jpg|jpeg|gif|bmp|ico|png|swf)$ {
access_log off;
}
}

```

Besagte Konfigurationsdatei mit dem namen proxy-config wird im folgenden Verzeichnis abgelegt:

/etc/nginx/sites-available

```

:/etc/nginx/sites-available# touch proxy-config
:/etc/nginx/sites-available# nano proxy-config
:/etc/nginx/sites-available# █

```

Um diese zu aktivieren muss die aktuelle konfiguration aus demsites-enabled Ordner gelöscht und durch die neue ersetzt werden:

```


root@kmu-Acadia:/etc/nginx/sites-available# rm ../sites-enabled/default
root@kmu-Acadia:/etc/nginx/sites-available# ln -s ../sites-enabled/proxy-config proxy-config
ln: failed to create symbolic link 'proxy-config': File exists
root@kmu-Acadia:/etc/nginx/sites-available# ln -s proxy-config ../sites-enabled/proxy-config
root@kmu-Acadia:/etc/nginx/sites-available# ls -la ../sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 Mar 21 11:30 .
drwxr-xr-x 6 root root 4096 Mar 21 11:02 ..
lrwxrwxrwx 1 root root  12 Mar 21 11:30 proxy-config -> proxy-config
root@kmu-Acadia:/etc/nginx/sites-available# █

```

Danach wird der NGINX Service neugestartet.

Jetzt sind die einzelnen Webdienste erreichbar.

← → ↻ acadia.schreib.at ☆ ⓘ 🔒 font ⓘ B



# Webmin

You must enter a username and password to login to the server on acadia.schreib.at

Remember me

## Einrichten von NextCloud



Create an admin account

root

cisoclass 

So-so password

Storage & database ▾

Data folder

/var/www/nextcloud/data

Configure the database

Only MySQL/MariaDB is available. Install and activate additional PHP modules to choose other database types.

For more details check out the [documentation](#). ↗


root

cisoclass 

nextcloud

localhost

**Finish setup**

 Need help? [See the documentation](#) ↗

## Einrichten von GitLab

Das Passwort wird auf cisoclass gesetzt.

git.acadia.schreib.at/users/password/edit?reset\_password\_token=dK-nb6Bjxj\_Pqkgnk54y

## GitLab Community Edition

**Open source software to collaborate on code**

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

### Change your password

New password  
.....

Confirm new password  
.....

Change your password

Didn't receive a confirmation email? [Request a new one](#)

Already have login and password? [Sign in](#)

## Let's Encrypt Zertifikat Generieren

Die Let's Encrypt Zertifikate werden als Teil einer späteren Übung konfiguriert.

## VPN

Es wird pptpd installiert `apt-get install pptpd`

Konfigurationsdatei `etc/pptp.conf`:

```
option /etc/ppp/options.pptpd
localip 10.0.0.37 # local vpn IP
remoteip 10.0.0.200-204 # ip range for connection
```

Konfigurationsdatei `etc/ppp/options.pptpd`:

```
#custom settings for a simple fast pptp server
ms-dns 8.8.8.8
ms-dns 4.2.2.2
lock
name pptpd
require-mschap-v2
# Require MPPE 128-bit encryption
# (note that MPPE requires the use of MSCHAP-V2 during authentication)
require-mppe-128
```

[http://localhost:4000/NVS/5CHIF\\_20170321\\_Schreib/](http://localhost:4000/NVS/5CHIF_20170321_Schreib/)