

Software Defined Infrastructure

Daniel Hiller & Micha Huhn

July 30, 2021

Introduction

Contributing

License

DNS

Querying DNS data

Querying www.spotify.com

Installing Bind

Bibliography

LDAP

Recommended Preparations

Exercises

Apache Web Server

Exercises

File Cloud

Exercises

Introduction

Contributing

Contributing

Found an error or have a suggestion? Please open an issue on GitHub (github.com/dentremor/Software-Defined-Infrastrucure):



Figure 1: QR code to source repository

License

License



Figure 2: AGPL-3.0 license badge

Software Defined Infrastructure (c) 2021 Daniel Hiller and contributors

SPDX-License-Identifier: AGPL-3.0

DNS

Querying DNS data

Querying DNS data

Due to the absence of `dig`, this was installed with the following command:

```
$ apt install dnsutils
```

Querying www.hdm-stuttgart.de

MX:

```
$ dig +nocmd hdm-stuttgart.de mx +noall +answer:
```

```
hdm-stuttgart.de. 2752      IN  MX  10  mx2.hdm-stuttgart.de
hdm-stuttgart.de. 2752      IN  MX  10  mx4.hdm-stuttgart.de
hdm-stuttgart.de. 2752      IN  MX  10  mx3.hdm-stuttgart.de
hdm-stuttgart.de. 2752      IN  MX  10  mx1.hdm-stuttgart.de
```

```
$ dig +noall +answer 10 mx2.hdm-stuttgart.de.:
```

```
mx2.hdm-stuttgart.de. 3197    IN  A    141.62.1.23
```

```
$ dig +nocmd +noall +answer -x 141.62.1.23:
```

```
23.1.62.141.in-addr.arpa. 3142    IN  PTR  mx2.hdm-stuttgart.de
```

NS:

```
$ dig +nocmd hdm-stuttgart.de ns +noall +answer:
```

```
hdm-stuttgart.de. 3590      IN  NS  iz-net-4.hdm-stuttgart.de
hdm-stuttgart.de. 3590      IN  NS  iz-net-3.hdm-stuttgart.de
hdm-stuttgart.de. 3590      IN  NS  dns1.belwue.de.
hdm-stuttgart.de. 3590      IN  NS  iz-net-2.hdm-stuttgart.de
```

Querying www.spotify.com

Querying www.spotify.com

CNAME:

```
$ dig +noall +answer www.spotify.com:
```

```
www.spotify.com. 230 IN CNAME edge-web-split-geo.dual
```

```
edge-web-split-geo.dual-gslb.spotify.com. 80 IN A 35.18
```

```
$ dig +noall +answer -x 35.186.224.25:
```

```
25.224.186.35.in-addr.arpa. 120 IN PTR 25.224.186.35.1
```

Installing Bind

Installing Bind

With the following command we can install bind9 and bind9utils:

```
$ apt install bind9 bind9utils
```

In `/etc/bind/` we need to adjust the `named.conf.options`, for that we need to know the IP-address of our domain `sdi3a.mi.hdm-stuttgart.de` to which we want to forward. For that we can use the following command:

```
$ dig +nocmd sdi3a.mi.hdm-stuttgart.de +noall +answer:  
sdi3a.mi.hdm-stuttgart.de. 86400 IN A 141.62.75.103
```

Now we can enter the IP-address in the already mentioned file.

Configure the zone file

To register our zones (which we will create later) we need to adjust the file `:named.conf.local` which should look like the following:

```
//  
// Do any local configuration here  
//  
  
zone "mi.hdm-stuttgart.de" {  
  
    type master;  
  
    file "/etc/bind/zones/db.forward";  
  
    allow-transfer { 141.62.75.103; };  
  
};
```

```
zone "75.62.141.in-addr.arpa" {
```


Create cache directory

```
$ mkdir -p /var/cache/bind
```

Configure the created zones

In the first step we need to change our directory to

```
$ cd /etc/bind
```

```
$ mkdir zones
```

Configure forward zone

We start to configure our forward lookup zone `zones/db.forward` with

```
$ vim db.forward
```

To get the host record we need to dig `sdi3a.mi.hdm-stuttgart.de`.

```
$ dig +noall +answer sdi3a.mi.hdm-stuttgart.de.:
sdi3a.mi.hdm-stuttgart.de. 86400 IN  A  141.62.75.103
```

With this information we can adjust our file `zones/db.forward` which looks like the following:

```
; db.forward
; Forward lookup zone
```

```
$TTL 604800
```

```
@          IN          SOA          ns3.mi.hdm
          01;
          28800;
          7200;
```

Configure reverse zone

With the information we got from above through the dig command, we can configure our reverse zone:

```
; db.rev-local  
; reverse lookup zone
```

```
$TTL 604800
```

```
@           IN           SOA           ns3.mi.ho  
           01;  
           28800;  
           7200;  
           2419200;  
           86400;  
  
)  
  
           NS           ns3.  
103        IN           PTR           sdi3a.mi.l
```

Forwarders

We use the Cloudflare DNS service as a forwarder.

Add the forwarder in the file `/etc/bind/named.conf.options`:

```
forwarders {  
    1.1.1.1  
};
```

Set mail exchange record

To achieve this we need to set another record in our forward zone
etc/bind/zones/db.forward:

```
mail                IN                MX
```

Test the record via nslookup:

```
$ nslookup manual.sdi3a.mi.hdm-stuttgart.de 141.62.75.103
```

```
Server:      141.62.75.103
```

```
Address:     141.62.75.103#53
```

```
Name:   manual.sdi3a.mi.hdm-stuttgart.de
```

```
Address: 141.62.75.103
```

```
$ nslookup -type=ptr 141.62.75.103
```

```
Server:      127.0.0.53
```

```
Address:     127.0.0.53#53
```

```
103.75.62.141.in-addr.arpa name = sdi3a.mi.hdm-stuttgart.de
```

```
103.75.62.141.in-addr.arpa name = dh102.sdi3a.mi.hdm-stuttgart.de
```

Bibliography

LDAP

Recommended Preparations

What is the LDAP Protocol? What is the difference between the two protocols LDAP and LDAPS?

“The Lightweight Directory Access Protocol can be used for querying and modifying information from distributed directory services.”

The difference between these two protocols are the encryption. LDAPS is encrypted via SSL and running on the default port 636, LDAP is encrypted or decrypted via START TLS and running on the default port 389. (“Editorial - LDAP”, 2021)

What does the acronym dc in dc=somedomain, dc=org stand for?

It stands for domain component and represents the namespaces of an object (Willeke, 2019).

What is the role of LDAP ObjectClass definitions? How do they relate to LDAP schema definitions?

The ObjectClass is a LDAP Schema element AttributeType (Willeke, 2019).

Describe the relationship between LDAP entries and ObjectClass values.

Each LDAP entry in the Directory Information Tree has an ObjectClass attribute. The values of this attribute can be modified but not removed (Willeke, 2019).

Is it possible to dynamically change an entry's structure?

No, the structure must conform to the constraint defined by the LDAP Schema (Willeke, 2019).

What does the term “bind to an LDAP” server mean?
What is an “anonymous” bind?

Bind is used to authenticate clients to the directory server.

There are three elements included in the request:

1. LDAP protocol version
2. Distinguished Name (DN)
3. Credentials for user authentication

At an anonymous bind the above points 2. and 3. are submitted as an empty string.

(Wilson)

Do LDAP servers in general support database features like transactions, ACID semantic etc.?

“Lightweight Directory Access Protocol (LDAP) Transactions is defined in RFC 5805 and is defined as “Experimental”.

As with distinct update operations, each transaction has atomic, consistency, isolation, and durability properties ACID.” (Willeke, 2017)

Explain the term “replication” in an LDAP server context.

For distribution reasons the LDAP-database can be distributed to several servers. There exists one master on which write-operations are allowed. On the others you can only pull the changes from the master (Anonym, 2019).

Why do organizations sometimes prefer LDAP data repositories rather than using relational database systems?

LDAP is very suitable in cases of high read rates and low write rates (write-once-read-many-times). Furthermore, relational databases like SQL requires a detailed knowledge about the data structure, which isn't the case when it comes to LDAP. (ZyTrax, 2019)

How is the LDIF format being organized? Explain the practical use of LDIF data when running an LDAP service.

The format is organized with objects and attributes. The LDIF data describes the directory structure which is needed for exchange (“Editorial - LDIF”, 2021)

LDAP filters

How do LDAP filters work?

There are several filters in LDAP with which it is possible to add criteria to an object search. (Föckeler)

What is the meaning of the term scope?

The LDAP search scope indicates the set of entries at or below the BaseDN that may be considered potential matches for a SearchRequest (Willeke, 2019).

How do predicate based filters connected by logical and/or/not look like?

And: (& (...K1...) (...K2...) (...K3...) (...K4...)) Or: (| (...K1...) (...K2...) (...K3...) (...K4...)) Not: (! (...K1...) (...K2...) (...K3...) (...K4...))

OpenLDAP server software specific questions

What does the term “database backend” refer to with respect to OpenLDAP server implementation?

Backend does the actual work of storing or retrieving data in response to LDAP requests. Backend may be compiled statically into `slapd`, or when module support is enabled, they may be dynamically loaded (Open LDAP Foundation, 2021).

Why is LDAP replication important?

The risk of a failure will be minimized, and the traffic load will be reduced.

Bibliography

Willeke, J. (various dates). LDAP Wiki 3. May 2021, from <https://ldapwiki.com/wiki>

Editorial - LDAP. (2021, April 19). In Wikipedia. https://de.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

Editorial - LDIF. (2021, April 19). In Wikipedia. https://de.wikipedia.org/wiki/LDAP_Data_Interchange_Format

Bosswell, W. (2003, October 10). ObjectClasses queried 3. May 2021, from <https://www.informit.com/articles/article.aspx?p=101405&seqNum=7#>

Wilson, N. (No datum available). The LDAP Bind Operation queried 3. May 2021, from <https://ldap.com/the-ldap-bind-operation/>

Anonym (2019, September 3). LDAP Wiki 3. May 2021, from <https://ldapwiki.com/wiki>

ZyTrax Inc. (2019, February 19). LDAP Concepts & Overview 7.

Exercises

Browse an existing LDAP Server

No Authentication vs. Authentication?

When you are authenticated on the LDAP-server, you can see all data which belongs to your user. When you are not authenticated, you can also see all data except the `matrikelNr`.

Set up an OpenLdap server

First we need to install several packages on our server:

```
$ apt install slapd ldap-utils dialog
```

To reconfigure slapd we need to type into our console:

```
$ dpkg-reconfigure slapd
```

```
DNS-Domainname: sdi3a.mi.hdm-stuttgart.de
```

Populating your DIT

After adding all entries in our tree, it looks like the following:

```
version: 1
```

```
dn: dc=betrayer,dc=com
```

```
objectClass: dcObject
```

```
objectClass: organization
```

```
objectClass: top
```

```
dc: betrayer
```

```
o: betrayer.com
```

```
dn: cn=admin,dc=betrayer,dc=com
```

```
objectClass: organizationalRole
```

```
objectClass: simpleSecurityObject
```

```
cn: admin
```

```
userPassword:: e1NTSEF9UUpzZm96RVFxVTFadEhGN3VrWE96dDNZRi9H
```

```
description: LDAP administrator
```

```
dn: ou=departments,dc=betrayer,dc=com
```

Testing a bind operation as a non - admin user

The image displays two side-by-side screenshots of the Apache Directory Studio interface, illustrating the configuration for testing a bind operation as a non-admin user.

Left Screenshot: LDAP Browser View
The LDAP browser shows the tree structure of the LDAP directory. The selected entry is `uid=frida,ou=financial,ou=departments,dc=be`. The attribute description table is as follows:

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	person (structural)
objectClass	top (abstract)
cn	frida.smith
sn	smith
uid	frida

The "Properties for 'D. Bean'" dialog is open, showing the "Connection" tab. The "Bind DN or user:" field is set to `uid=daniel_s_ndevel,ou=software,ou=departments,dc=be`, which is circled in red. The "Authentication Method" is "Simple Authentication" and the "Authorization ID (SASL)" is "SASL_PLAIN only".

Right Screenshot: LDAP Browser View
The LDAP browser shows the tree structure of the LDAP directory. The selected entry is `uid=frida,ou=financial,ou=departments,dc=be`. The attribute description table is as follows:

Attribute Description	Value
objectClass	inetOrgPerson (structural)
objectClass	organizationalPerson (structural)
objectClass	person (structural)
objectClass	top (abstract)
cn	frida.smith
sn	smith
uid	frida

The "Properties for 'admin@sd3a.mh...dm-stuttgart.de'" dialog is open, showing the "Connection" tab. The "Bind DN or user:" field is set to `cn=admin,dc=be,ou=financial,ou=departments,dc=be`, which is circled in red. The "Authentication Method" is "Simple Authentication" and the "Authorization ID (SASL)" is "SASL_PLAIN only".

Filter based search

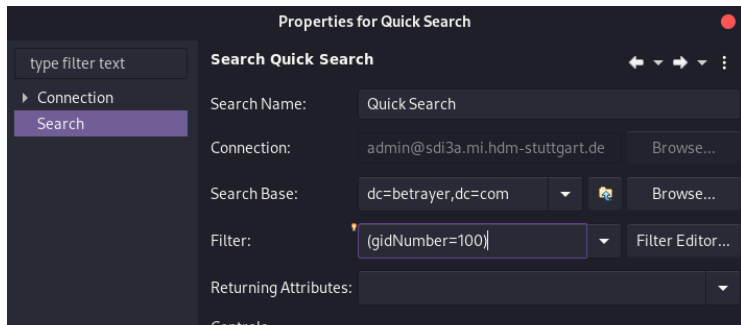
All users with a uid attribute value starting with the letter “b”:

`(uid=b*)`

All entries with either a defined uid attribute or a ou attribute starting with the letter “d”:

`(|(uid=d*)(ou=d*))`

All users entries within the whole DIT having a gidNumber value of 100:



The screenshot shows a window titled "Properties for Quick Search" with a dark background. On the left is a sidebar with a search input field containing "type filter text" and two menu items: "Connection" and "Search" (which is highlighted in purple). The main area is titled "Search Quick Search" and contains several fields: "Search Name" is "Quick Search"; "Connection" is "admin@sdi3a.mi.hdm-stuttgart.de" with a "Browse..." button; "Search Base" is "dc=betrayer,dc=com" with a dropdown arrow and a "Browse..." button; "Filter" is "(gidNumber=100)" with a dropdown arrow and a "Filter Editor..." button; "Returning Attributes" is empty with a dropdown arrow. At the bottom, there are "Controls" and "Help" buttons.

Extending an existing entry

The entry

`uid=bean,ou=devel,ou=software,ou=departments,dc=betrayer;dc=`
can be extended by the `objectclass=posixAccount`. Construct
an LDIF file to add the attributes `uidNumber`, `gidNumber` and
`homeDirectory` by a `modify/add` operation:

```
uid=bean, ou=devel, ou=software, ou=departments, dc=betrayer;dc=  
changetype: add  
objectClass: posixAccount  
uidNumber: 42  
gidNumber: 1337  
homeDirectory: /home/daniel
```

Accessing LDAP data by a mail client

The screenshot shows a software interface for an address book. The main window, titled "Address Book", has a menu bar with "File", "Edit", "View", "Tools", and "Help". Below the menu is a toolbar with icons for "New Contact", "New List", "Edit", "Write", and "Delete", along with a search box labeled "Name or Email".

The main area displays a table of contacts with columns: "Name", "Email", "Chat Name", "Organization", and "Work Phone". The "Name" column contains icons of a person and the names: "admin", "betrayder.com", "D. Bean", "D. Smith", "F. Bean", "F. Smith", "T. Bean", and "T. Smith". The "Email" column shows "betrayder.com" for "D. Bean". The "Organization" column shows "betrayder.com" for "D. Bean".

An "Edit Contact for D. Bean" dialog box is open in the foreground. It has tabs for "Contact", "Private", "Work", "Other", "Chat", and "Photo". The "Contact" tab is active. The dialog contains the following fields and options:

- First: [Empty text box]
- Last: [Text box containing "Bean"]
- Display: [Text box containing "D. Bean"]
- Always prefer display name over message header
- Nickname: [Empty text box]
- Email: [Empty text box]
- Additional Email: [Empty text box]
- Chat Name: [Empty text box]
- Work: [Empty text box]
- Home: [Empty text box]
- Fax: [Empty text box]
- Pager: [Empty text box]
- Mobile: [Empty text box]
- Prefer to receive messages formatted as: [Dropdown menu showing "Unknown"]

At the bottom of the dialog are "Cancel" and "OK" buttons.

LDAP configuration

The screenshot shows the LDAP Browser interface. The left pane displays a tree view with 'DIT' expanded to 'Root DSE (2)' and 'dc:'. The right pane shows the 'Properties for "admin@sdi3a.mi...dm-stuttgart.de"'. The 'Connection' tab is active, showing the following configuration:

- Authentication Method: Simple Authentication
- Authentication Parameter: Bind DN or user: cn=admin, dc=betrayer, dc=com
- Authorization ID (SASL): SASL PLAIN only
- Bind password: [masked]
- Save password
- Buttons: Check Authentication, SASL Settings, Kerberos Settings

A 'Check Authentication' dialog box is overlaid on the bottom, displaying an information icon and the message: 'The authentication was successful.' with an 'OK' button.

LDAP based user login

Test connection to active directory

Use the following command:

```
root@sdi3b:~# telnet sdi3a.mi.hdm-stuttgart.de 389
```

Then something like this should appear:

```
Trying 141.62.75.103...
```

```
Connected to sdi3a.mi.hdm-stuttgart.de.
```

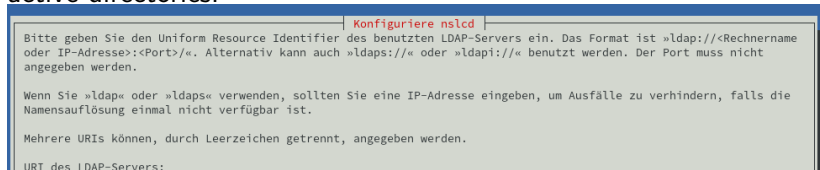
```
Escape character is '^]'.  
^C
```

Install and configure libpam-ldapd

```
$ apt-get install libpam-ldapd
```

After the installation, a window will open where we can configure the package.

In the following window we need to enter the hostname to our active directories.



Backup and recovery / restore

Create a backup of the OpenLDAP database configuration in a LDIF-file.

```
$ slapcat -b cn=config -l ldap-config.ldif
```

Create a backup of the OpenLDAP data.

```
$ slapcat -l ldap-data.ldif
```

Copy the data and configuration backup from the OpenLDAP provider server to the OpenLDAP consumer server.

```
$ scp {ldap-data.ldif,ldap-config.ldif} root@sdi3b.mi.hdm-s
```

Now we need to access our consumer server via ssh.

```
$ ssh root@sdi3b.mi.hdm-stuttgart.de
```

Restore the OpenLDAP provider Data and configs on the consumer server. Stop the LDAP service:

```
$ systemctl stop slapd
```

Accessing LDAP by a Python application.

Please find the application and the associated README.md in the Python directory.

```
[danny@localhost Python]$ make run USER=dh102
pip install -r ldaper/requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: ldap3==2.9 in /home/danny/.local/lib/python3.9/site-packages (from -r ldaper/requirements
.txt (line 1)) (2.9)
Requirement already satisfied: click==8.0.1 in /home/danny/.local/lib/python3.9/site-packages (from -r ldaper/requiremen
ts.txt (line 2)) (8.0.1)
Requirement already satisfied: pyasn1>=0.4.6 in /home/danny/.local/lib/python3.9/site-packages (from ldap3==2.9->-r ldap
er/requirements.txt (line 1)) (0.4.8)
python3 ldaper/cli.py dh102
Password:
Repeat for confirmation:
----- Results -----
version: 1
dn: uid=dh102,ou=userlist,dc=hdm-stuttgart,dc=de
objectClass: hdmAccount
objectClass: hdmStudent
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
objectClass: eduPerson
uid: dh102
mail: dh102@hdm-Stuttgart.de
uidNumber: 67954
cn: Hiller Daniel
loginShell: /bin/sh
hdmCategory: 1
gidNumber: 100
homeDirectory: /home/stud/d/dh102
sn: Hiller

# total number of entries: 1
```

The following frameworks are used:

▶ <https://www.python-ldap.org/en/python-ldap-3.3.0/>

▶ <https://click.palletsprojects.com/en/8.0.x/>

Apache Web Server

Exercises

First Steps

For the following tasks we need the package Apache2, which we can install with the following command:

```
$ aptitude install apache2
```

After we install the package, Apache is running per default and in our case it can be queried with `http://sdi3a.mi.hdm-stuttgart.de/`.

When we move the `index.html` file out of the directory, we can discover another page. For this we need to query the address again. Now we can see an empty table and below that we find the version of our Apache Server, the domain where it is hosted and the associated port.

In the next step we provide our own simple webpage which looks like the following:

```
<!DOCTYPE html>
```

```
<html>
```

```
  <body>
```

Virtual hosts

To realize virtual hosts we need to create a .con file in /etc/apache2/sites-available. The config in this file should look like the following:

```
<VirtualHost *:80>
    ServerAdmin dh102@hdm-stuttgart.de
    ServerName sdi3a.mi.hdm-stuttgart.de
    ServerAlias dh102.sdi3a.mi.hdm-stuttgart.de
    DocumentRoot /home/sdidoc/
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Now the site must be enabled with:

```
$ a2ensite dh102.conf
```

Add the following instructions to /etc/apache2/apache2.conf:

```
<Directory /home/sdidoc/>
    AllowOverride None
```

SSL / TLS Support

First, we need to create our private root key with a bit length of 2048:

```
$ openssl genrsa -out rootCA.key 2048
```

For security reasons we should encrypt our key:

```
$ openssl genrsa -des3 -out rootCA.key 2048
```

With our rootCA.key we can now self-sign this certificate:

```
$ openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 365
```

The above command starts an interactive script, which in our case looked like the following after processing:

```
You are about to be asked to enter information that will be
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished
```

```
There are quite a few fields but you can leave some blank
```

```
For some fields there will be a default value,
```

```
If you enter ' ' the field will be left blank
```

LDAP authentication

For these exercises we use our user "daniel" from 2.2.9 LDAP based user login.

To use LDAP with Apache Web Server, we need to enable the module authnz_ldap:

```
$ a2enmod authnz_ldap
```

We can copy one of our previous .conf files and edit the config, which should look like the following:

```
<VirtualHost *:443>
    ServerAdmin dh102@hdm-stuttgart.de
    DocumentRoot /home/sdidoc/
    SSLEngine on
    SSLCertificateFile "/root/ssl-cert/device.crt"
    SSLCertificateKeyFile "/root/ssl-cert/device.key"
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
<Directory "/home/sdidoc">
```


Mysql™ database administration

To install mysql-server use:

```
$ apt install default-mysql-server
```

After facing an issue with LXC Container we need to adjust the config /etc/systemd/system/mariadb.service.d/lxc.conf:

```
[Service]
```

```
ProtectHome=false
```

```
ProtectSystem=false
```

```
# These settings turned out to not be necessary in my case.
```

```
#PrivateTmp=false
```

```
#PrivateNetwork=false
```

```
PrivateDevices=false
```

and run the following commands:

```
$ systemctl daemon-reload
```

```
$ systemctl restart mariadb
```

Providing WEB based user management to your LDAP Server

To install the LDAP Account Manager we need to download it and forward it to the server via scp because ldap-account-manager isn't available via the official apt repositories:

```
https://sourceforge.net/projects/lam/
```

```
$ scp /home/user/Downloads/ldap-account-manager_7.6-1_all.d
```

... and install it with apt:

```
$ apt install /home/ldap-account-manager_7.6-1_all.deb
```

Now we can configure the LDAP Account Manager

```
http://sdi3a.mi.hdm-stuttgart.de/lam/templates/config/index
```

The default master password for Edit general settings is lam and should be changed to something secure.

The password for Edit server profiles is also lam. Here we can edit TLS and a List of valid users:

Publish your documentation

Our documentation is written as a .md file, so we need to convert it with Pandoc into a valid .html file:

```
$ docker run -v "${PWD}:/data:z" pandoc/latex doku.md --num
```

Transfer the .html file to our server with scp:

```
$ scp index.html root@sdi3a.mi.hdm-stuttgart.de:/home/sdidoc
```

We don't use rsync because we need to convert our file with Pandoc anyway to get an actual version. But if you want to use rsync the command would be:

```
$ rsync -avz -e ssh root@sdi3a.mi.hdm-stuttgart.de:/home/sd
```

We can adjust the .conf file etc/apache2/apache2.conf/ and add:

```
<Directory /home/sdidoc/>  
    AllowOverride None  
    Require all granted  
    Options Indexes FollowSymLinks
```

File Cloud

Exercises

Setup Nextcloud with Apache Web Server

First, we need to install packages for Apache2, MariaDB and PHP:

```
$ apt install vim unzip
$ apt install apache2 mariadb-server libapache2-mod-php
$ apt install php-gd php-json php-mysql php-curl
$ apt install php-intl php-mcrypt php-imagick
$ apt install php-zip php-xmlwriter php-xmlreader php-xml p
```

We need another user for our Nextcloud in our database:

```
$ mariadb
> CREATE USER 'ncadmin'@'localhost' IDENTIFIED BY 'test1';
> CREATE DATABASE IF NOT EXISTS nextcloud CHARACTER SET utf8;
> GRANT ALL PRIVILEGES ON nextcloud.* TO 'ncadmin'@'localhost';
> FLUSH PRIVILEGES;
> quit;
```

In the next step we download Nextcloud and move it to
/var/www:

```
$ wget https://download.nextcloud.com/server/releases/latest
```

User authentication with LDAP

To enable LDAP support click on Icon in the right top corner and navigate to Apps:

