

Praktikumsbericht zum “Aufbau eines Clusters”

IM WISE 2017 BEI DR. MICHAEL KUHN

MARVIN HEUER

Inhaltsverzeichnis

1 Einführung und Motivation	4
2 Grundinstallation, Netzwerk, NFS	4
2.1 Einführung	4
2.2 Installation	5
2.2.1 Installation der virtuellen Maschinen	5
2.2.2 Interneteinrichtung	7
2.2.3 NFS Installation	8
2.3 Aufgetretene Probleme.....	10
3 Benutzerverwaltung	10
3.1 Motivation	10
3.2 Installation von OpenLDAP	11
3.2.1 Einrichtung.....	11
3.2.2 Benutzer hinzufügen	14
3.2.3 Clienteinrichtung	15
3.3 Aufgetretene Probleme.....	15
4 Konfigurationsverwaltung.....	16
4.1 Motivation	16
4.2 Installation von Ansible	16
4.3 Aufgetretene Probleme.....	17
5 Monitoring	17
5.1 Motivation	17
5.2 Installation	17
5.2.1 Einrichtung von Prometheus.....	17
5.2.2 Installation von Grafana.....	19
5.3 Aufgetretene Probleme.....	20
6 Scheduling	20
6.1 Motivation	20
6.2 Installation von Slurm	21
6.3 Aufgetretene Probleme.....	24
7 Softwareverwaltung	24
7.1 Motivation	24
7.2 Installation von EasyBuild	25
7.3 Aufgetretene Probleme.....	25
Literatur- und Quellenverzeichnis	27

1 Einführung und Motivation

Ich zitiere von der Veranstaltungsseite: „Unter einem Parallelrechner versteht man einen Rechner, auf dem Anwendungen echt parallel ausgeführt werden können. Die heutzutage üblichen Desktop-PCs sind bereits kleine Parallelrechner, da mehrere Kerne eine echt parallele Abarbeitung von Anwendungen erlauben. Vor allem aufgrund der Verbreitung von letzteren wird in Zukunft die Programmierung von parallelen Programmen und Algorithmen an Bedeutung gewinnen. Cluster kombinieren mehrere Rechner über eine schnelle Vernetzung, um auch größere Probleme schnell lösen zu können. Existierende Cluster-Supercomputer kommen auf mehrere 100.000 Prozessoren.“¹

Cluster benötigt man im Rahmen des Hochleistungsrechnen. Dabei werden Cluster genutzt um die Berechnung, Modellierung und Simulation komplexer Systeme und die Verarbeitung riesiger Messdatenmengen zu unterstützen.

„High Performance Computing-Cluster, die oft auch als Superrechner bezeichnet werden. Ziel einer derartigen Konstruktion ist es, extrem anspruchsvolle Rechenaufgaben in einzelne Bestandteile zu zerlegen, auf die Knoten zu verteilen und dort gleichzeitig ausführen zu lassen. Zu diesem Zweck ist ein extrem schnelles Netzwerk vonnöten, so dass hier oft InfiniBand zum Einsatz kommt. Die meisten der auf einem HPC-Cluster basierenden Supercomputer werden derzeit unter Linux betrieben.“²

2 Grundinstallation, Netzwerk, NFS

2.1 Einführung

Der erste Schritt zum Aufbau eines Cluster ist die Einrichtung von sogenannten *Backend*-Servern.³ Im Rahmen des Praktikums wurden dazu virtuelle Maschinen in Virtual Box eingerichtet. Auf diesen Maschinen wurde CentOS 7 installiert und eingerichtet. CentOS ist eine Linuxdistribution, die auf *Red Hat Linux* aufbaut⁴ und prädestiniert ist für Serveranwendungen, wie es hier der Fall ist.⁵ Damit ein Cluster koordiniert Aufgaben verarbeiten kann, bedarf es einem lokalen Netzwerk, welches die *Backend*-Server mit einem zentralen Loginknoten verbindet, von dem aus ein Loginserver bzw. ein Scheduler

1 Arbeitsgruppe Wissenschaftliches Rechnen: „Praktikum ‚Aufbau eines Clusters‘ – Beschreibung“, URL: https://wr.informatik.uni-hamburg.de/teaching/wintersemester_2017_2018/aufbau_eines_clusters, Stand: 26.03.2018

2 IT-Administrator.de: „Cluster“, URL: http://www.it-administrator.de/themen/server_client/grundlagen/172792.html, Stand: 26.03.2018

3 Chef: „Chef Documentation“, URL: https://docs.chef.io/install_server_ha.html, Stand: 26.03.2018

4 Tecmint.com: „RedHat vs Debian : Administrative Point of View“, URL: <https://www.tecmint.com/redhat-vs-debian-administrative-point-of-view/>, Stand: 26.03.2018

5 The CentOS Project: „About CentOS“, URL: <https://www.centos.org/about/>, Stand: 26.03.2018

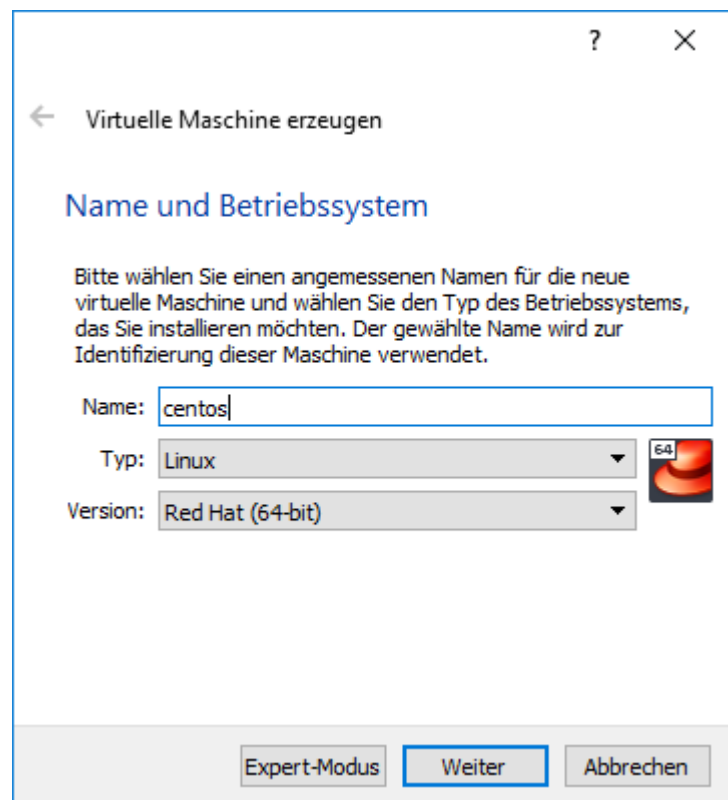
arbeiten kann. Das Netzwerk wird mittels Konfiguration der hosts-Datei und Konfiguration in Virtual Box eingerichtet. Die hosts-Datei ist eine lokale Konfigurationsdatei, um Hostnamen zu IP-Adressen zuzuordnen.⁶ In einem kleinen Cluster ist dies noch händisch machbar, bei größeren Clustern wird dazu ein DNS-Server konfiguriert, z.B. via Dnsmasq.⁷ Ein Network File System (NFS) erlaubt das Verwalten von Dateien auf mehreren Computern innerhalb eines Netzwerkes so, als wären sie auf der lokalen Festplatte gespeichert. Dadurch ist es nicht nötig zu wissen, wo die Dateien physikalisch gespeichert sind, um auf sie zuzugreifen und ein Nutzer kann sich auf den verschiedenen Servern anmelden und auch immer auf seine Daten zugreifen.⁸

2.2 Installation

2.2.1 Installation der virtuellen Maschinen

Um mit der Installation beginnen zu können, müssen erstmal ein paar Voraussetzungen geschaffen werden. Zuerst muss Virtual Box heruntergeladen und installiert werden. Virtual Box ist eine Virtualisierungssoftware. Damit können wir virtuelle PCs erstellen und betreiben. Des Weiteren benötigen wir ein DVD-Image (*.ISO-Datei, nicht die minimale, sondern die DVD-Variante(!)) von CentOS 7, welches wir hier als Betriebssystem nutzen.

Es ist darauf zu beachten, dass man nun einheitlich eine 32bit- oder 64bit-Version herunterlädt und in Virtual Box einstellt. Um eine virtuelle Maschine zu erzeugen, öffnen wir Virtual Box und klicken auf den Button „Neu“ bzw. „New“. Als Namen geben wir nun „Centos“ ein. Die darunterliegenden Felder Typ und Version sollten sich nun automatisch auf „Linux“ und „Red Hat[...]“ ändern. Falls sie dies nicht tun, so ist der Typ auf „Linux“ und die Version auf „Red Hat[...]“ zu ändern. Jetzt klicken wir auf den Knopf „Weiter“. Das nächste

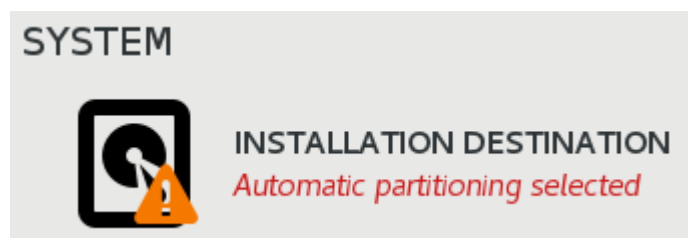


⁶ Rheinwerk Verlag: Die Datei /etc/hosts, URL: http://openbook.rheinwerk-verlag.de/unix_guru/node202.html, Stand: 26.03.2018

⁷ Ubuntuusers: Dnsmasq, URL: <https://wiki.ubuntuusers.de/Dnsmasq/>, Stand: 26.03.2018

⁸ Selflinux: NFS – Network File System, URL: <http://www.selflinux.org/selflinux/html/nfs.html>, Stand: 26.03.2018

Fenster heißt „Speichergröße“. Die empfohlene Größe beträgt 1024 MB und genau so lassen wir auch die Einstellung und drücken auf „Weiter“. Der letzte Schritt ist die „Platte“, also erstmal die Entscheidung ob eine Festplatte erzeugt werden soll oder eine vorhandene verwendet wird. Wir erzeugen eine neue, dies ist auch die Standardeinstellung. Wir drücken auf den Button „Erzeugen“. Es öffnet sich ein neues Fenster zur Erzeugung der virtuellen Festplatte. Wir lassen alles auf den Standardeinstellungen bis auf die Größe der Festplatte. Diese legen wir auf 25 GB fest, bei kleineren Größen kann es zu sehr un schönen Fehler kommen, die u.a. zum fatalen Systemabsturz führten. Wir klicken auf „Erzeugen“ und unsere Maschine taucht in Virtual Box in der linken Liste der Maschinen mit dem Status „ausgeschaltet“ auf. Nun starten wir mit der eigentlichen Installation von CentOS. Dazu wählen wir unsere soeben erstellte Maschine aus und drücken auf „Starten“. Es öffnet sich ein neues Fenster, welches mit „centos [wird ausgeführt]“ überschrieben ist. Jetzt gibt es zwei Möglichkeiten, entweder es wird schon vorgeschlagen ein Medium auszuwählen oder in der Konsole steht die Meldung „FATAL: No bootable medium found! System halted.“. Letzterer Fehler entsteht, da noch kein Medium eingelegt worden ist. Unter der Konsole ist eine Statusleiste mit einer kleinen CD als Logo. Darauf machen wir nun einen Rechtsklick und drücken in dem aufploppenden Kontextmenü „Abbild auswählen...“. In dem auftauchenden Dateisystem navigieren wir zu unserem CentOS-Image und wählen dies aus. Nun schließen wir unsere Konsole mit einem Klick auf das rote Kreuz oben rechts. Virtual Box fragt nun, ob wir „den Zustand der virtuellen Maschine speichern“ oder „die virtuelle Maschine ausschalten“ möchten. Wir entscheiden uns für letzteres und quittieren mit einem Klick auf „OK“. Jetzt starten wir unsere Maschine wieder. In der Konsole wählen wir jetzt mit den Pfeiltasten „Install CentOS 7“ und bestätigen mit Enter. Nach einiger Zeit öffnet sich nun der „WELCOME TO CENTOS 7.“-Installationsbildschirm. Die Sprache lassen wir auf Englisch und drücken auf „Continue“. Wir kommen nun zu einer allgemeinen Übersicht, wo man bei Bedarf das Tastaturlayout, Zeit oder Sprache noch ändern könnte. In jedem Fall muss ein Klick auf „Installation destination“ gemacht



werden, da dies mit einem gelben Ausrufezeichen markiert ist. Allerdings ist dort schon alles korrekt eingestellt, sodass wir oben direkt auf „Done“ wieder klicken. Zurück in der Übersicht können wir auf „Begin Installation“ klicken und es geht los. In den „user settings“ gehen wir auf „Root password“, um dies zu setzen. Hier geben wir zweimal das gewünschte root-Passwort ein. Dies ist sehr wichtig und sollte man sich notieren

(sicherheitstechnisch schlecht) oder sich gut merken. Durch einen Klick auf „Done“ endet der Prozess und wir warten auf das Ende der Installation.

Am Ende müssen wir einmal auf „Reboot“ klicken, um CentOS nutzen zu können. Im Loginbildschirm loggen wir uns mit dem Namen „root“ und dem selbst festgelegten Passwort. Wir haben CentOS erfolgreich installiert.

2.2.2 Interneteinrichtung

Wir fahren nun unsere Maschine wieder runter, um das Internet zu aktivieren. Dazu müssen wir in die globalen Einstellungen von Virtual Box mit der Tastenkombination „Strg+G“. Im Tab „Netzwerk“ können wir ein NAT-Netzwerk anlegen. In dem Tab sehen wir am rechten Rand im Fenster eine Netzwerkkarte mit einem grünen Plus und drücken darauf. Nun schließen wir wieder unsere Einstellungen mit einem Klick auf „OK“ (!).

Allerdings müssen wir noch dafür sorgen, dass auch unsere Maschine dieses Netzwerk nutzt. In unserer Liste der installierten Maschinen machen wir einen Rechtsklick und im Kontextmenü klicken auf „Ändern“. Es öffnet sich das maschinenspezifische Einstellungsfenster. Auch hier gehen wir auf den Tab „Netzwerk“. Bis jetzt ist unter Adapter 1 ausgewählt: „Angeschlossen an: NAT“. Dies ändern wir auf „NAT-Netzwerk“ durch einen Klick auf die Liste. Ein NAT-Netzwerk ist eine Mischung aus den Einstellungen „NAT“ und „Internes

Netzwerk“. Die virtuellen Maschinen in demselben Netzwerk können

untereinander und mit dem externen Netzwerk kommunizieren. Des Weiteren aktivieren

wir einen zweiten Netzwerkkarte, den wir auf „Host-only“-Network stellen. Jetzt können wir wieder unsere Maschine starten und uns einloggen. Wenn wir nun testen mittels ping-Befehl ob unser Internet geht, werden wir feststellen, dass dieses noch nicht geht. Wir müssen nämlich unsere Verbindung im Netzwerkmanager aktivieren. Dazu geben wir in die Konsole den Befehl `nmtui` ein. Es öffnet sich eine Benutzeroberfläche in unserem Terminal, in dem wir auf die Option „Activate a connection“. Im nachfolgenden Fenster drücken wir für unsere Internetverbindung entsprechend „<Activate>“. Einmal drücken wir



„<Back>“. Jetzt haben wir Internet, allerdings nur bis wir die Maschine neustarten. Deshalb müssen wir nochmal auf die Option „Edit a connection“. Dort editieren wir mittels Auswahl und „<Edit...>“ die Netzwerkverbindung und setzen ein Kreuz bei „Automatically connect“.

Alle in 2.2 genannten Schritte müssen nun noch wiederholt werden bis die gewünschte Anzahl an

Nodes erstellt ist.

Eine unserer Maschinen bestimmen wir nun zum „Serverknoten“, dieser stellt also den Loginserver und den NFS Server. Bevor wir weitermachen, müssen wir auf allen virtuellen Maschinen einmal die hosts-Datei anpassen. In den hosts-Dateien legen wir fest, wie die lokalen IP-Adressen und Namen aufgelöst werden sollen. Wichtig ist, dass die hosts-Datei komplett identisch ist auf allen Maschinen. Die hosts-Datei finden wir in `/etc/`. Die Datei können wir mit einem Editor freier Wahl anpassen. Als Editor ist vi z.B. schon vorinstalliert.

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
10.0.2.4 node1.dkrz.local node1
10.0.2.3 master.dkrz.local master
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
```

2.2.3 NFS Installation⁹

2.2.3.1 Servereinrichtung

Zuerst präparieren wir unseren Serverknoten. Erstmal installieren wir mittels `yum install nfs-utils` ein wichtiges Paket zur NFS-Erstellung. Anschließend erstellen wir ein Verzeichnis, welches wir mittels NFS teilen wollen (→ `mkdir /var/nfsshare`). Zusätzlich müssen wir die Rechte anpassen mit folgenden Befehlen:

⁹ Howtoforge.com: NFS Server and Client Installation on CentOS 7, URL: <https://www.howtoforge.com/nfs-server-and-client-on-centos-7>, Stand: 26.03.2018

```
chmod -R 755 /var/nfsshare  
chown nfsnobody:nfsnobody /var/nfsshare
```

Es müssen einige Dienste gestartet werden und zum Starten beim Booten eingestellt werden. Dazu geben wir die folgenden Befehle nacheinander ein:

```
systemctl enable rpcbind  
systemctl enable nfs-server  
systemctl enable nfs-lock  
systemctl enable nfs-idmap  
systemctl start rpcbind  
systemctl start nfs-server  
systemctl start nfs-lock  
systemctl start nfs-idmap
```

Um das NFS-Verzeichnis jetzt im Netzwerk verfügbar zu machen, müssen wir die exports-Datei anpassen, dabei wollen wir zwei Punkte definieren. Die Datei finden wir im etc-Ordner. Mittels **vi /etc/exports** öffnen wir die Datei und editieren sie so:

```
/var/nfsshare *(rw,sync,no_root_squash,no_all_squash)  
/home *(rw,sync,no_root_squash,no_all_squash)
```

Um den NFS-Service neuzustarten, geben wir **systemctl restart nfs-server** ein und schalten die Firewall aus. Hauptsächlich um die nervige Ausnahmereingabe zu umgehen, die uns noch häufiger antreffen wird. Die Firewall schaltet man mittels **systemctl disable firewalld** und **systemctl stop firewalld** aus.

2.2.3.2 Clienteinrichtung

Dies geschieht nun alles auf unseren Clients. Zu Beginn installieren wir die NFS Utilities mit **yum install nfs-utils**. Im Anschluss erstellen wir die Verzeichnisse in welche gemountet werden sollen. Dazu geben wir die folgenden beiden Befehle ein:

```
mkdir -p /mnt/nfs/home  
mkdir -p /mnt/nfs/var/nfsshare
```

Jetzt mounten wir das NFS-geteilte Homeverzeichnis im Client:

```
mount -t nfs 10.0.2.4:/home /mnt/nfs/home/
```

Dadurch mounten wir das /home des NFS-Servers. Nun machen wir dasselbe für das andere Verzeichnis:

```
mount -t nfs 10.0.2.4:/var/nfsshare /mnt/nfs/var/nfsshare/
```

Das Ganze ist jetzt allerdings bei einem Neustart wieder verloren, deshalb müssen wir das noch permanent festlegen:

Mit **vim /etc/fstab** öffnen wir den Editor und editieren rein:

[...]

```
10.0.2.4:/home /mnt/nfs/home nfs defaults 0 0
```

```
10.0.2.4:/var/nfsshare /mnt/nfs/var/nfsshare nfs defaults 0 0
```

Damit erreichen wir permanent erhaltene Mountpunkte. Die in 2.2.3.2 beschriebenen Schritte müssen selbstredend auf allen Clients erfolgen.

2.3 Aufgetretene Probleme

Der Umstieg von Windows auf Linux stellte mich zu Beginn vor ein großes Problem. Ich habe mich in Linux eingearbeitet und dabei eine Menge gelernt. Ein Problem stellt sich dar, falls bei der Installation von CentOS lokale Benutzer angelegt werden. Diese müssen dann dem LDAP-Server hinzugefügt werden und die Synchronisierung der persönlichen Daten sichergestellt werden.

3 Benutzerverwaltung

3.1 Motivation

Eine zentrale Benutzerverwaltung ist erstrebenswert in einem Cluster, da somit ein Account auf allen Knoten verwendet werden kann. In Verbindung mit NFS kann man somit auf allen Teilen des Clusters sich einloggen und auf seine Daten zugreifen. Es gibt verschiedene Implementationen, die in einem Cluster benutzt werden können wie z.B. Active Directory, LDAP, FreeIPA, NIS oder Samba. Grundsätzlich sind diese relativ ähnlich, manche stellen noch weitergehende Services zur Verfügung. Im Rahmen des Praktikums nutze ich LDAP mit der Implementierung OpenLDAP, wobei in einem kleinen Netzwerk wie hier, auch NIS verwendet werden könnte. LDAP ist von seinem Umfang und seiner Flexibilität her eher für größere Umgebungen gedacht.

3.2 Installation von OpenLDAP¹⁰

3.2.1 Einrichtung

Auf unserem Loginknoten müssen wir zunächst OpenLDAP installieren. Dazu führen wir den folgenden Befehl aus:

```
yum -y install openldap compat-openldap openldap-clients openldap-servers  
openldap-servers-sql openldap-devel
```

Damit haben wir die benötigten LDAP-Pakete installiert. Nun starten wir noch den LDAP Service und aktivieren den Autostart des Servers:

```
systemctl start slapd.service  
systemctl enable slapd.service
```

Jetzt legen wir ein Rootpasswort für LDAP fest. Dies sollten wir uns notieren und nicht vergessen mit dem Befehl **slappasswd**.

Anschließend konfigurieren wir unseren OpenLDAP Server. Dazu navigieren wir in das Verzeichnis `/etc/openldap/slapd.d/` und erstellen dort eine *.ldif-Datei mittels bspw. **vi db.ldif** . Wir füllen die Datei in folgender Weise:

```
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
replace: olcSuffix  
olcSuffix: dc=dkrz,dc=local
```

```
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
replace: olcRootDN  
olcRootDN: cn=ldapadm,dc=dkrz,dc=local
```

```
dn: olcDatabase={2}hdb,cn=config  
changetype: modify  
replace: olcRootPW  
olcRootPW: *****
```

¹⁰ IT'zGeek: Step By Step OpenLDAP Server Configuration on CentOS 7 / RHEL 7, URL: <https://www.itzgeek.com/how-tos/linux/centos-how-tos/step-step-openldap-server-configuration-centos-7-rhel-7.html>, Stand: 26.03.2018

Sobald die Datei erstellt wurde, müssen wir diese Konfigurationsdatei an den LDAP Server senden mittels **ldapmodify -Y EXTERNAL -H ldapi:/// -f db.ldif**

Jetzt passen wir die monitor.ldif an, sodass nur unser angelegter LDAP-root-Nutzer Zugriff hat. Dazu schreiben wir **vi monitor.ldif** in die Konsole. Wir editieren die Datei wie folgt:

```
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external, cn=auth" read by
dn.base="cn=ldapadm,dc=itzgeek,dc=local" read by * none
```

Anschließend übermitteln wir die Konfiguration an den Server durch:

```
ldapmodify -Y EXTERNAL -H ldapi:/// -f monitor.ldif
```

Nun generieren wir ein LDAP Certificate durch:

```
openssl req -new -x509 -nodes -out /etc/openldap/certs/itzgeekldapcert.pem -keyout
```

Das Ergebnis wird im Verzeichnis **/etc/openldap/certs/** abgelegt. Bei der Erzeugung werden wir nach ein paar Infos gefragt, welche die Erzeugung des Schlüssels beeinflussen. Im Anschluss daran setzen wir unsere Berechtigungen auf LDAP:

```
chown -R ldap:ldap /etc/openldap/certs/*.pem
```

Wir müssen jetzt eine Konfigurationsdatei anlegen, damit LDAP eine sichere Verbindung über unsere Zertifikate nutzt. Wie gehabt erzeugen wir dazu eine Datei mit **vi certs.ldif** :

```
dn: cn=config
changetype: modify
replace: olcTLSCertificateFile
olcTLSCertificateFile: /etc/openldap/certs/dkrzldapcert.pem
```

```
dn: cn=config
changetype: modify
replace: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/openldap/certs/dkrzldapkey.pem
```

Abschließend laden wir für den Server die Konfigurationsdatei hoch:

```
Idapmodify -Y EXTERNAL -H Idapi:/// -f certs.ldif
```

Jetzt kommen wir zur Datenbankkonfiguration. Dazu kopieren wir die Beispielformatdateien in das Verzeichnis `/var/lib/ldap` und aktualisieren danach die Dateizugriffsrechte.

```
cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG  
chown ldap:ldap /var/lib/ldap/*
```

Nun fügen wir verschiedene Schemata hinzu durch:

```
Idapadd -Y EXTERNAL -H Idapi:/// -f /etc/openldap/schema/cosine.ldif
```

```
Idapadd -Y EXTERNAL -H Idapi:/// -f /etc/openldap/schema/nis.ldif
```

```
Idapadd -Y EXTERNAL -H Idapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

Anschließend erstellen wir eine Domaindatei namens `base.ldif` wie folgt:

```
dn: dc=dkrz,dc=local
```

```
dc: dkrz
```

```
objectClass: top
```

```
objectClass: domain
```

```
dn: cn=ldapadm ,dc=dkrz,dc=local
```

```
objectClass: organizationalRole
```

```
cn: ldapadm
```

```
description: LDAP Manager
```

```
dn: ou=People,dc=dkrz,dc=local
```

```
objectClass: organizationalUnit
```

```
ou: People
```

```
dn: ou=Group,dc=dkrz,dc=local
```

```
objectClass: organizationalUnit
```

```
ou: Group
```

Nun können wir die Verzeichnisstruktur generieren.

```
Idapadd -x -W -D "cn=ldapadm,dc=dkrz,dc=local" -f base.ldif
```

3.2.2 Benutzer hinzufügen

Um einen Benutzer dem LDAP Server hinzufügen, muss zuerst eine LDIF-Datei erstellt werden.

Beispielhaft für den Benutzer pasja: **vi pasja.ldif**

```
dn: uid=pasja,ou=People,dc=dkrz,dc=local  
objectClass: top  
objectClass: account  
objectClass: posixAccount  
objectClass: shadowAccount  
cn: pasja  
uid: pasja  
uidNumber: 9999  
gidNumber: 100  
homeDirectory: /home/pasja  
loginShell: /bin/bash  
gecos: pasja  
userPassword: pasja123  
shadowLastChange: 17058  
shadowMin: 0  
shadowMax: 99999  
shadowWarning: 7
```

Nun fügen wir die angelegte Datei in den Server ein mit dem Befehl Idapadd:

```
Idapadd -x -W -D "cn=ldapadm,dc=dkrz,dc=local" -f pasja.ldif
```

Wir bestätigen die Eingabe mit dem LDAP Adminpasswort. Anschließend setzen wir ein Passwort:

```
Idappasswd -s pasja123 -W -D "cn=ldapadm,dc=dkrz,dc=local" -x  
"uid=pasja,ou=People,dc=dkrz,dc=local"
```

Mit dem LDAP Search-Befehl kann man nach LDAP Accounts suchen.

Bsp.: **ldapsearch -x cn=pasja -b dc=dkrz,dc=local**

Alternativ kann man mit Ldapdelete Einträge löschen:

**ldapdelete -W -D "cn=ldapadm,dc=dkrz,dc=local"
"uid=pasja,ou=People,dc=dkrz,dc=local"**

Das Bearbeiten der Firewall können wir hier überspringen.

3.2.3 Clienteinrichtung

Auf der Clientseite haben wir deutlich weniger zu tun je Maschine. Als erstes installieren auf dem Client die benötigten LDAP-Tools mit:

yum install -y openldap-clients nss-pam-ldapd

Als nächstes fügen wir die Clientmaschine zum LDAP Server. Die grüne Adresse muss durch die IP-Adresse des LDAP Servers ersetzt werden:

**authconfig --enableldap --enableldapauth --ldapserver=10.0.2.4 --
ldapbasedn="dc=dkrz,dc=local" --enablemkhomedir --update**

Und danach starten wir den LDAP Clientservice neu.

systemctl restart nslcd

Damit ist der Client fertig eingerichtet. Diese Schritte von 3.2.3 müssen nun für alle Clients wiederholt werden.

3.3 Aufgetretene Probleme

Mit dieser Anleitung sind die häufigsten Probleme auf die man trifft Tippfehler. Da mir dies selber ein paar Mal passiert ist, der dringliche Hinweis nochmal drüber zu schauen, dass keine „Typer“ auftreten.

4 Konfigurationsverwaltung

4.1 Motivation

Eine Software-Konfigurationsverwaltung gibt uns die Möglichkeit auf all unseren Clusterrechnern schnell und zügig Softwarekonfigurationen zu ändern. Auch hier treten wieder Skaleneffekte auf, d.h. bei einer größeren Clientzahl spielen die Systeme ihre Stärken aus. Ich habe mich hier im Rahmen des Praktikums für Ansible entschieden. In Ansible werden Playbooks angelegt. Diese beschreiben welche Konfiguration auf den Clients installiert werden soll. Im Gegensatz zu anderen Konfigurationsverwaltungen setzt Ansible dabei auf eine Push-Strategie, d.h. die Clients fragen nicht regelmäßig den Server nach Updates, sondern der Server „pusht“ einfach seine neuen Playbooks. Das ganze geschieht dabei über SSH.

4.2 Installation von Ansible¹¹

Die Installation beginnt mit mit den folgenden beiden Befehlen:

```
sudo yum install epel-release  
sudo yum install ansible
```

Eine wichtige Voraussetzung ist, dass alle Computer mit denen man durch Ansible konfigurieren will, mit `ssh root@unsere_client_ip` kommunizieren können ohne nach einem Passwort gefragt zu werden. Dazu erzeugen wir auf unserem Loginserver einen SSH-Key mit `ssh-keygen` und kopieren diesen anschließend mit `ssh -i ~/.ssh/id_rsa root@unsere_client_ip` auf unseren Client. Dies tun wir für alle Clienten.

Jetzt haben wir die Voraussetzungen geschaffen, um loslegen zu können.

Zunächst bearbeiten wir die Liste der Ansible Hosts. Dazu öffnen wir z.B. mit `vi /etc/ansible/hosts` selbstverständlich mit root-Rechten.

Diese Datei enthält sehr viele Kommentare, am Ende fügen wir folgendes ein:

```
[servers]
```

```
node1 ansible_ssh_host=10.0.2.15  
node2 ansible_ssh_host=10.0.2.16
```

¹¹ Red Hat, Inc.: Ansible Documentation – Installation Guide, URL: http://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html, Stand: 27.03.2018

Danach speichern wir die Datei. Wir bringen jetzt Ansible bei, dass er die soeben konfigurierten Server mit einem speziellen Benutzer ansprechen soll. Zunächst erzeugen wir dafür das Verzeichnis `group_vars` mit **`mkdir /etc/ansible/group_vars`**

In diesem Ordner erzeugen wir Dateien für jede Gruppe, die wir konfigurieren.

Mittels `nano /etc/ansible/group_vars/servers` erzeugen wir eine neue Datei und fügen dieser folgenden Code hinzu:

`ansible_ssh_user: root`

Im Nachhinein müssen noch Playbooks entsprechend den Bedürfnissen des Nutzers angelegt werden.

4.3 Aufgetretene Probleme

Die Einrichtung von Ansible ging problemlos und ziemlich „straight forward“. Dabei sind keine Probleme aufgetreten. Wichtig ist es die Playbooks richtig anzulegen, dafür sei an andere Stelle verwiesen.

5 Monitoring

5.1 Motivation

Durch ein attraktives Monitoring können unsere Clusterrechner überwacht werden. Dies kann sinnvoll sein, um die CPU-Auslastung und allgemeine Leistungswerte unserer Maschinen bei rechenintensiven Aufgaben im Auge zu behalten. Es gibt verschiedene Software wie z.B. Munin, Ganglia oder Prometheus. Im Rahmen des Praktikums nutze ich Prometheus in Verbindung mit Graphana.

5.2 Installation

5.2.1 Einrichtung von Prometheus

Zuerst bereiten wir ein neues Verzeichnis vor namens Downloads, wo wir alle Downloads abspeichern. Das heißt wir führen die beiden Befehle aus:

`mkdir ~/Downloads`

`cd ~/Downloads`

Mit curl können wir die neueste Version von Prometheus runterladen.

curl -LO

"https://github.com/prometheus/prometheus/releases/download/v2.2.1/prometheus-2.2.1.linux-amd64.tar.gz"

Jetzt legen wir ein neues Verzeichnis namens Prometheus an.

mkdir ~/Prometheus

cd ~/Prometheus

In dieses neu angelegte Verzeichnis entpacken wir unser soeben heruntergeladenes *.tar.gz-Archiv mittels

tar -xvzf ~/Downloads/prometheus-2.2.1.linux-amd64.tar.gz

Der nächste Schritt ist das Installieren von Node Exporter. Diesen brauchen wir um die Daten unseres CentOS-Servers in ein für Prometheus lesbares Format zu konvertieren.

Wir navigieren in unser Downloads-Verzeichnis und nutzen **curl** um Node Exporter runterzuladen:

cd ~/Downloads && curl -LO

"https://github.com/prometheus/node_exporter/releases/download/v0.15.2/node_exporter-0.15.2.linux-amd64.tar.gz"

Nun erstellen wir ein neues Verzeichnis für den Node Exporter, navigieren dort hin und entpacken da das heruntergeladene Archiv.

mkdir ~/Prometheus/node_exporter

cd ~/Prometheus/node_exporter

tar -xvzf ~/Downloads/node_exporter-0.15.2.linux-amd64.tar.gz

Indem wir nun in das Node Exporter Verzeichnis navigieren, können wir mittels **.node_exporter** den Node Exporter starten.

Um nun den Prometheusserver selbst zu starten, müssen wir erst noch eine Konfigurationsdatei anlegen. Also navigieren wir erstmal in unser Prometheusverzeichnis:

```
cd ~/Prometheus/prometheus-2.2.1.linux-amd64
```

Dort erstellen wir mittels **vi ~/Prometheus/prometheus-2.2.1.linux-amd64/prometheus.yml** die Konfigurationsdatei und fügen folgenden Code in die Datei:

```
scrape_configs:
```

```
- job_name: "node"
  scrape_interval: "15s"
  static_configs:
    - targets: ['localhost:9090']
```

Mit **./prometheus** starten wir den Prometheusserver.

5.2.2 Installation von Grafana

Nun installieren wir Grafana. Dazu fügen wir dem Yum-Repository Grafana hinzu, indem wir folgendes ausführen:

```
vi /etc/yum.repos.d/grafana.repo
```

In diese Datei tragen wir folgendes ein:

```
[grafana]
```

```
name=grafana
baseurl=https://packagecloud.io/grafana/stable/el/6/$basearch
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://packagecloud.io/gpg.key
https://grafanarel.s3.amazonaws.com/RPM-GPG-KEY-grafana
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

Dadurch sind wir in der Lage einfach mit **yum install grafana** Grafana zu installieren.

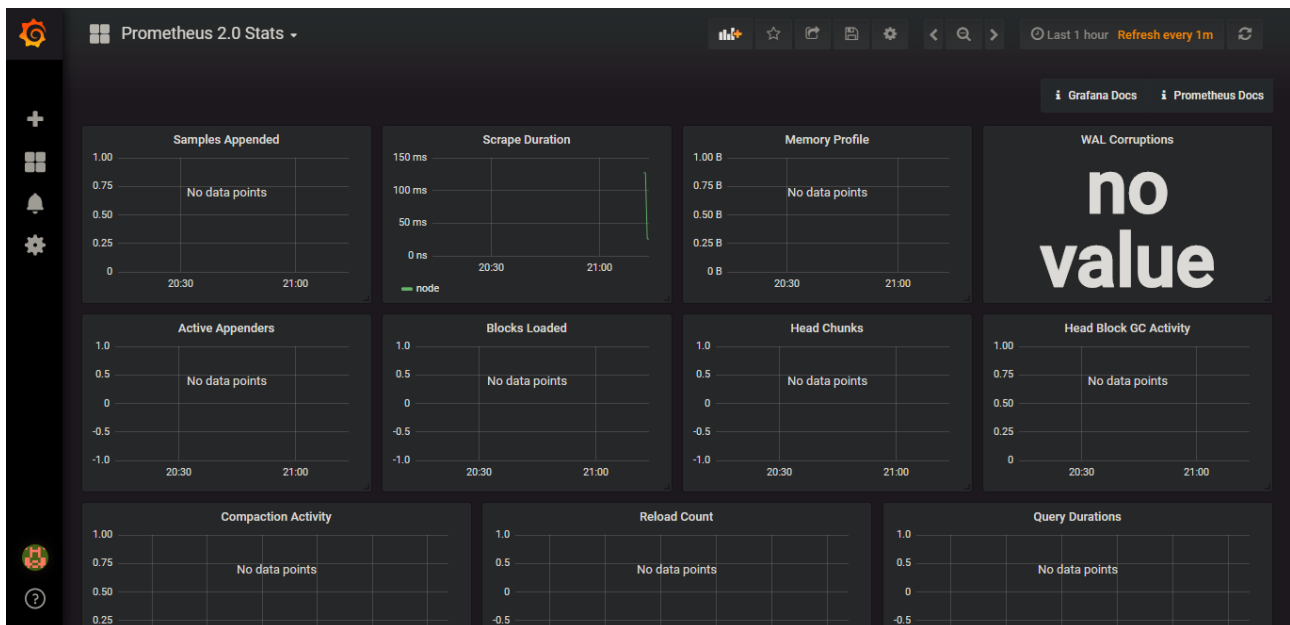
Mit den folgenden Befehlen starten wir den Grafanaserver:

systemctl daemon-reload

systemctl start grafana-server

Mit dem Befehl **systemctl status grafana-server** können wir überprüfen, ob der Grafanaserver läuft. Damit dieser voll funktioniert, müssen zeitgleich auch der Node Exporter und der Prometheusserver gestartet sein.

Auf Grafana zugreifen kann man dann mittels `https://unsere_ip:3000/`. Da brauchen wir Zugangsdaten. Die Standardzugangsdaten sind als Benutzername: admin und als Passwort: admin auch. In Grafana selbst müssen wir Prometheus als direkte Datenquelle einrichten. Des Weiteren können wir in Grafana weitere Dashboards importieren, die z.B. die CPU-Auslastung anzeigen o.Ä. Dies geht ziemlich einfach, über eine Import-ID, die eindeutig einem Dashboard zugewiesen ist.



5.3 Aufgetretene Probleme

Bei der Einrichtung von Grafana wird „leider“ standardmäßig keine direkte Datenquelle vorgeschlagen, sondern indirekt über einen Proxy. Dies ist leider eine sehr unglückliche Standardeinstellung, da dies in unserem Fall nicht von Vorteil ist.

6 Scheduling

6.1 Motivation

Mit einem Batch Scheduler können Batchaufträge im Cluster an die verteilten Ressourcen und angeschlossenen Backend-Server übermittelt werden. Dadurch wird eine hohe

Verlässlichkeit und Stabilität im Cluster erreicht. Ein Batch Scheduler verwaltet eine Jobschlange für das Cluster. Der Scheduler weist diese Jobs von der Schlange entsprechend den Ressourcen zu, wenn diese frei werden. Die Zuweisungen und ähnliches werden dabei im Hintergrund vollzogen. Es gibt verschiedenste Programme wie Slurm, Torque, Moab, OpenLava etc. Im Rahmen des Praktikums beschränke ich mich auf Slurm.

6.2 Installation von Slurm

Zunächst einmal sorgen wir für eine Datenbank um die Logs zu speichern. Wir nutzen dazu MariaDB. Dies installieren wir auf unserer Servernode:

```
yum install mariadb-server mariadb-devel -y
```

Bevor wir weitermachen, geben wir folgenden Code einmal auf jeder Maschine ein, um einheitliche Accounts auf allen Nodes zu erhalten. Die Raute markiert den Anfang der Befehle und muss nicht eingegeben werden.

```
# export MUNGEUSER=991  
# groupadd -g $MUNGEUSER munge  
# useradd -m -c "MUNGE Uid 'N' Gid Emporium" -d /var/lib/munge -u $MUNGEUSER  
-g munge -s /sbin/nologin munge  
# export SLURMUSER=992  
# groupadd -g $SLURMUSER slurm  
# useradd -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g  
slurm -s /bin/bash slurm
```

Jetzt können wir mit der eigentlichen Installation von Munge und Slurm beginnen. Wir beginnen mit Munge. Dazu installieren wir die folgenden beiden Programme:

```
yum install epel-release  
yum install munge munge-libs munge-devel -y
```

Nach dem Installieren von Munge müssen wir auf unserem Server einen geheimen Schlüssel erzeugen. Dies bereiten wir durch folgendes vor:

```
yum install rng-tools -y  
rngd -r /dev/urandom
```

Das eigentliche Erstellen passiert durch:

```
/usr/sbin/create-munge-key -r
```

```
dd if=/dev/urandom bs=1 count=1024 > /etc/munge/munge.key
```

```
chown munge: /etc/munge/munge.key
```

```
chmod 400 /etc/munge/munge.key
```

Nachdem der geheime Schlüssel erstellt wurde, müssen wir diesen an alle „Clients“ übermitteln mit scp:

```
scp /etc/munge/munge.key root@node1:/etc/munge
```

```
scp /etc/munge/munge.key root@node2:/etc/munge
```

Jetzt verbinden wir uns mittels SSH in unsere „Nodes“. Auf jeder unserer Nodes korrigieren wir nun unsere Berechtigungen:

```
chown -R munge: /etc/munge/ /var/log/munge/
```

```
chmod 0700 /etc/munge/ /var/log/munge/
```

Und starten Munge neben den Nodes auch auf dem Server mit:

```
systemctl enable munge
```

```
systemctl start munge
```

Nun kommen wir zu Slurm selbst auf unserem Server. Slurm benötigt einige Pakete um vernünftig funktionieren zu können. Deshalb geben wir folgenden Befehl ein:

```
yum install openssl openssl-devel pam-devel perl-devel numactl numactl-devel
```

```
hwloc hwloc-devel lua lua-devel readline-devel rrdtool-devel ncurses-devel
```

```
man2html libibmad libibumad -y
```

Jetzt laden wir Slurm herunter mit wget.

```
mkdir /Downloads
```

```
cd /Downloads
```

```
wget https://download.schedmd.com/slurm/slurm-17.11.5.tar.bz2
```

Anschließend installieren wir RPMBuild:

```
yum install rpm-build  
rpmbuild -ta slurm-17.11.5.tar.bz2
```

Dadurch erhalten wir einen Build von Slurm, den wir nun auf ins Verzeichnis packen, da dieses synchronisiert wird (siehe NFS). Wir kopieren mit cp:

```
mkdir /nfs/slurm-rpms
```

```
cp slurm-17.11.5-1.el7.centos.x86_64.rpm slurm-devel-17.11.5-1.el7.centos.x86_64.rpm slurm-munge-17.11.5-1.el7.centos.x86_64.rpm slurm-perlapi-17.11.5-1.el7.centos.x86_64.rpm slurm-plugins-17.11.5-1.el7.centos.x86_64.rpm slurm-sjobexit-17.11.5-1.el7.centos.x86_64.rpm slurm-sjstat-17.11.5-1.el7.centos.x86_64.rpm slurm-torque-17.11.5-1.el7.centos.x86_64.rpm  
/nfs/slurm-rpms
```

Dann installieren wir auf jeder Maschine unseren Clusters Slurm:

```
yum --nogpgcheck localinstall slurm-17.11.5-1.el7.centos.x86_64.rpm slurm-devel-17.11.5-1.el7.centos.x86_64.rpm slurm-munge-17.11.5-1.el7.centos.x86_64.rpm slurm-perlapi-17.11.5-1.el7.centos.x86_64.rpm slurm-plugins-17.11.5-1.el7.centos.x86_64.rpm slurm-sjobexit-17.11.5-1.el7.centos.x86_64.rpm slurm-sjstat-17.11.5-1.el7.centos.x86_64.rpm slurm-torque-17.11.5-1.el7.centos.x86_64.rpm
```

Nach der Installation müssen wir eine Konfigurationsdatei anlegen. In einem Editor unserer Wahl legen wir nun eine slurm.conf an (Bsp.: **vim slurm.conf**). Dazu verwenden wir den Link: <https://slurm.schedmd.com/configurator.easy.html> . Dieses Tool wird von Slurm angeboten und empfohlen um seine Konfigurationsdatei zu erzeugen. Ans Ende der Konfigurationsdatei müssen noch Spezifikationen der Nodes rangefügt werden:

```
NodeName=node1 NodeAddr=node1_ip CPUs=1 State=UNKNOWN  
NodeName=node2 NodeAddr= node2_ip CPUs=1 State=UNKNOWN
```

Diese Konfigurationsdatei muss dann auf alle Nodes mittels scp kopiert werden. Bsp.:

```
scp slurm.conf root@node1/etc/slurm/slurm.conf
```

Auf Seiten des Servers müssen wir nun die Berechtigungen richtig setzen:


```
mkdir /var/spool/slurmctld  
chown slurm: /var/spool/slurmctld  
chmod 755 /var/spool/slurmctld  
touch /var/log/slurmctld.log  
chown slurm: /var/log/slurmctld.log  
touch /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log  
chown slurm: /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
```

Auf Seiten unserer Nodes müssen ebenfalls Berechtigungen angepasst werden:

```
mkdir /var/spool/slurmd  
chown slurm: /var/spool/slurmd  
chmod 755 /var/spool/slurmd  
touch /var/log/slurmd.log  
chown slurm: /var/log/slurmd.log
```

Auf unseren Clients führen mit den folgenden Befehlen Slurm aus:

```
systemctl enable slurmd.service  
systemctl start slurmd.service  
systemctl status slurmd.service
```

Auf unserem Server starten wir Slurm nun mit

```
systemctl enable slurmctld.service  
systemctl start slurmctld.service  
systemctl status slurmctld.service
```

6.3 Aufgetretene Probleme

Im Rahmen der Einrichtung werden Schlüssel erzeugt, die übertragen werden müssen, was relativ aufwendig ist. Sehr gut hingegen ist das Online-Konfigurationstool das von Slurm zur Verfügung gestellt wird.

7 Softwareverwaltung

7.1 Motivation

Mittels einer strukturierten Softwareverwaltung können die Nodes effizient mit aktueller Software ausgestattet werden. Durch Skaleneffekte wird die Funktion sehr spannend.

Während bei einer Node das ganze noch händisch gemacht werden kann, so ist bei hundert Nodes das ganze deutlich aufwendiger. Es gibt hierzu verschiedenste Software, ich beschränke mich auf EasyBuild.

7.2 Installation von EasyBuild¹²

Bevor wir beginnen, müssen wir die Voraussetzungen erfüllen, die laut Dokumentation vorgegeben werden. Das heißt, wir müssen Pip und Lmod installieren. Dies machen wir folgt:

```
yum install Lmod
```

```
curl -LO https://bootstrap.pypa.io/get-pip.py
```

```
python get-pip.py
```

```
python -m pip install --upgrade pip setuptools wheel
```

Jetzt können wir die Bootstrappingmechanik starten, indem wir zuerst die Python-Datei runterladen.

```
curl -LO https://raw.githubusercontent.com/easybuilders/easybuild-framework/develop/easybuild/scripts/bootstrap_eb.py
```

```
mkdir /easybuild
```

```
su pasja
```

```
python bootstrap_eb.py $HOME/EasyBuild
```

```
pip install --install-option "--prefix=$HOME/EasyBuild" easybuild
```

```
module use $EASYBUILD_PREFIX/modules/all
```

```
module load EasyBuild
```

Und damit ist Easybuild auch schon installiert. Mit **eb --version** können wir prüfen, ob EasyBuild auch funktioniert.

7.3 Aufgetretene Probleme

Grundsätzlich gibt sich Easybuild alle Mühe die Installation so reibungslos wie möglich ablaufen zu lassen. Sofern man die Voraussetzungen erfüllt, startet das Pythonskript zügig. Eine Einschränkung, die man erst bei der Ausführung selbst erfährt, ist, dass man kein root-Nutzer sein darf. Die Hilfe der Dokumentation, bei Installationsproblemen, sich an den Support zu wenden, ist nur wenig sinnvoll.

¹² Ghent University: Installing EasyBuild, URL: <http://easybuild.readthedocs.io/en/latest/Installation.html>, Stand: 29.03.2018

Literatur- und Quellenverzeichnis

- Chef: „Chef Documentation“, URL: https://docs.chef.io/install_server_ha.html, Abruf am 26.03.2018.
- Ghent University: “Installing EasyBuild”, URL: <http://easybuild.readthedocs.io/en/latest/Installation.html>, Abruf am 29.03.2018.
- Howtoforge.com: “NFS Server and Client Installation on CentOS 7”, URL: <https://www.howtoforge.com/nfs-server-and-client-on-centos-7>, Abruf am 26.03.2018.
- IT‘zGeek: “Step By Step OpenLDAP Server Configuration on CentOS 7 / RHEL 7”, URL: <https://www.itzgeek.com/how-tos/linux/centos-how-tos/step-step-openldap-server-configuration-centos-7-rhel-7.html>, Abruf am 26.03.2018.
- Red Hat, Inc.: “Ansible Documentation – Installation Guide”, URL: http://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html, Abruf am 27.03.2018.
- Rheinwerk Verlag: „Die Datei /etc/hosts“, URL: http://openbook.rheinwerk-verlag.de/unix_guru/node202.html, Abruf am 26.03.2018.
- Selflinux: „NFS – Network File System“, URL: <http://www.selflinux.org/selflinux/html/nfs.html>, Abruf am 26.03.2018.
- Tecmint.com: “RedHat vs Debian : Administrative Point of View”, URL: <https://www.tecmint.com/redhat-vs-debian-administrative-point-of-view/>, Abruf am 26.03.2018.
- The CentOS Project: “About CentOS”, URL: <https://www.centos.org/about/>, Abruf am 26.03.2018.
- Ubuntuusers: „Dnsmasq“, URL: <https://wiki.ubuntuusers.de/Dnsmasq/>, Abruf am 26.03.2018.
- Arbeitsgruppe Wissenschaftliches Rechnen: „Praktikum ‚Aufbau eines Clusters‘ – Beschreibung“, URL: https://wr.informatik.uni-hamburg.de/teaching/wintersemester_2017_2018/aufbau_eines_clusters, Abruf am 26.03.2018.
- IT-Administrator.de: „Cluster“, URL: http://www.it-administrator.de/themen/server_client/grundlagen/172792.html, Abruf am 26.03.2018.