

Der Cumulus-Praktiker

Datacenter-Switching mit Linux

Markus Stubbig

Inhaltsverzeichnis

Vorwort

I. Für Einsteiger

1 Das Labornetzwerk

- Ressourcen
- Virtualisierung
- Hardware
- Netze
- Switches
- Adressierung
- Labor-Server
- Verwendung

2 Installation

- Vorbereitung
- VMware
- VirtualBox
- Hardware

3 Erste Schritte

- Bedienung
- Ablauf

Hilfe
Ersteinrichtung
Weitere Einrichtung
Repository
Lizenz
Zusammenfassung

4 Konnektivität

Routing
Nachbarschaftserkennung

II. Für Fortgeschrittene

5 Monitoring

Logging
SNMP
Kommandozeile
Prescriptive Topology
Zusammenfassung

6 Management Interface

Virtual Routing and Forwarding
Dienste
Kommandos
Technischer Hintergrund
Zusammenfassung

7 Switchports

Erste Schritte

Switching
Layer-3-Ports
Beispiel
Zusammenfassung

8 Ausfallschutz

Link Aggregation
Multi-Chassis Link Aggregation
Doppel-MLAG
Einrichtung
Virtual Router Redundancy
Zusammenfassung

9 Zentrale Authentifizierung

Protokolle
Laboraufbau
Microsoft Server
LDAP
RADIUS
TACACS+
Sicherheit
Technischer Hintergrund
Application Programming Interface
Zusammenfassung

10 Konfiguration

Ablauf
Archiv und Revision
Manuelles Backup

Automatisches Backup
Wiederherstellung
Zusammenfassung

11 Systemverwaltung

Updates
Rollback
Installation
Versionierung und Support
Zusammenfassung

III. Für Experten

12 sFlow

Inhalt einer Probe
Labor
Exporter
Kollektor
Automatische Konfiguration
Fehlersuche
Technischer Hintergrund
NetFlow
Zusammenfassung

13 Rapid Spanning-Tree

Crashkurs
Laboraufbau
Konfiguration
Best Practice

Technischer Hintergrund
Zusammenfassung

14 Spine/Leaf Topologie

Traditioneller Aufbau
Moderne Architektur
Was macht Cumulus?
Zusammenfassung

15 Dynamisches Routing

OSPF gegen BGP
OSPF
Skalierung
OSPFv3
Fehlersuche
Praxistest
Ausblick
Technischer Hintergrund
Zusammenfassung

16 Routing on the Host

Labor
Installation
Konfiguration
Erreichbarkeit
vRouter
Lastverteilung per Anycast
Sicherheit
Ausblick

Zusammenfassung

17 VXLAN

Grundlagen

Laboraufbau

Konfiguration

Lightweight Network Virtualization

Technischer Hintergrund

Ausblick

Zusammenfassung

18 Firewall

Cumulus Linux als Firewall

Laboraufbau

Paketfilter

Aktionen

Voreingestellte Regeln

Grenzen

Schutz für die Control-Plane

Technischer Hintergrund

Fehlersuche

Zusammenfassung

IV. Für Praktiker

19 Server

Laboraufbau

Linux

Windows Server

VMware ESXi
Zusammenfassung

20 Massenbereitstellung

Betriebssystem
Konfiguration
Zusammenfassung

21 Fehlersuche im Netz

cl-support
Netzadapter
Hardware überwachen
Logbuch
Kleine Werkzeuge
Zusammenfassung

22 Durchsatz messen

Auslastung
Durchsatzmessung
Zusammenfassung

V. Für Trickser

23 Best Practice

Änderungen mit Sicherungsnetz
Factory-Default
SSH-Login ohne Passworteingabe
Passwort zurücksetzen

24 Application Programming Interface

Wie funktioniert die API?

Lesender Zugriff

Schreibender Zugriff

API-Browser

Sicherheit

Technischer Hintergrund

Fehlersuche

Erweiterungen

Zusammenfassung

25 Life Hacks

Zugriff von Windows

NCLU erweitern

Web-API erweitern

Telegram

Cisco Discovery Protocol

Mausezahn

ACL-Ressourcen überwachen

Literaturverzeichnis

Stichwortverzeichnis

A. **Editor unter Linux**

B. **ASIC**

C. **Ausblick**

Ansible

Docker

Richtlinienbasiertes Routing

Virtual Routing and Forwarding

Multicast

Border Gateway Protocol

D. **Zusatzmaterial**

Vorwort

Ein weiteres Betriebssystem erblickt das Licht der Welt. Noch eins? In der Welt der Netzwerker scheint jeder Hersteller sein eigenes *Network-OS* zu kochen. Das Spektrum der Ergebnisse reicht von vielseitig über kurzlebig, verwaist bis zu stabil und weltbekannt. Manche Distributionen beginnen als Fork und enden nur Monate später.

Was ist also so toll an Cumulus Linux? Ganz einfach: Es ist eine Software ohne Hardware. Cumulus Linux ist ein *Bring your own Device* für Netzwerkschwitches. Hersteller-OS runter und Cumulus Linux drauf. Letztendlich ist ein Switch auch nur ein kleiner Server mit erstaunlich vielen Netzadaptern.

Das klingt nach Bastelbude und dem akademischen Wunsch, ein „falsches“ Betriebssystem auf einem ausgemusterten Gerät zu betreiben.

Cumulus Networks verfolgt diese Vorgehensweise als Geschäftsmodell. Also mit Anleitungen, Support und Eigenentwicklungen. Das Unternehmen produziert keine Hardware, sondern arbeitet mit Netzwerkausrüstern zusammen. Damit der Deckel perfekt auf den Topf passt, erhält Cumulus Einblick in die Hardwarespezifikation. Mit diesen Infos kommen die ASICs so richtig in Fahrt und treiben die Durchsatzraten in die Höhe.

Das Konzept hat leider ein operatives Problem: Für Netzwerker ist Cumulus zu viel Linux, Konfigurationsdateien und Skripte. Für Linux-Admins ist Cumulus zu viel Netzwerkereie, Protokolle und Adressen. Zum Glück schließt sich diese Lücke langsam durch fertige Bündel aus Hard-

und Software, gute Dokumentation, viele Erfolgsrezepte und ein hervorragendes Buch.

Viel Spaß beim Ausprobieren, Staunen und Fluchen.

Übersicht

Teil 1, *Für Einsteiger*, beginnt mit dem Aufbau der Netzwerk-Umgebung mit physikalischen Geräten oder auf einer virtuellen Plattform. Die erstellten Maschinen erhalten ihr Betriebssystem und eine erste Konfiguration. Anschließend gesellen sich die grundlegenden Funktionen Routing und IPv6 dazu.

In Teil 2, *Für Fortgeschrittene*, bekommen die Switches ernsthafte Aufgaben, die in jedem Netzwerk erfüllt sein müssen. Dabei zeigen die Switchports ihre Tricks bei VLANs, Trunks und ihre Flexibilität beim Bündeln von mehreren Leitungen.

Teil 3, *Für Experten*, taucht in Enterprise-Themen ein und beleuchtet den Einsatz im Rechenzentrum. Cumulus Linux prahlt mit seinen Fähigkeiten bei Lastverteilung, Ausfallschutz und dynamischem Routing, um die Infrastruktur fast unbesiegbar zu machen. Für tiefere Einsicht in die Masse der Datenverbindungen ist sFlow im Gepäck. Und die Switches können sogar einfache Aufgaben einer Firewall übernehmen.

In Teil 4, *Für Praktiker*, macht Cumulus Linux eine gute Figur beim Einsatz in großen Umgebungen durch unkomplizierte Installation und Änderung. Danach kommen die Server ans Netz: redundant, hochverfügbar oder als Teil der Routingdomäne.

Teil 5, *Für Trickser*, zeigt viele kleine Handgriffe, die die tägliche Arbeit mit Cumulus Linux reibungsfreier gestalten.

Und zuletzt kommt die Programmierschnittstelle von Cumulus Linux auf den Prüfstand.

Ressourcen

<https://cumulusnetworks.com>

Die Homepage von Cumulus Networks liefert einen guten Einstieg ins Thema und verlinkt zur Dokumentation, zum Forum und zum Download-Bereich.

<https://github.com/cumulusnetworks>

Die Entwickler hosten den Programmcode bei GitHub, wo jeder Einblick in den Fortschritt hat und sich an den Quellen bedienen kann. Daneben gibt es viele Demo-Projekte und Beispiele zum selber bauen.

<https://forums.cumulusnetworks.com>

Das Forum ist die erste Anlaufstelle für kleine Tutorials, Ideen, Diskussionen und Support aus der Community.

Schriftkonventionen

Nichtproportionalschrift zeigt die erzeugte Ausgabe eines Kommandos.

Schreibmaschinenschrift wird für Konfigurationen und Schlüsselwörter benutzt, die buchstabengetreu eingetippt werden müssen.

Nichtproportionalschrift Fett zeigt Befehle, die eine Ausgabe erwarten.

Hervorhebungen weisen auf besondere Wörter oder Zeilen innerhalb von Kommandos oder Bildschirmausgaben hin.

```
ein-sehr-langer-kommando-aufruf --mit --sehr \ --vielen  
"Optionen"
```

Kommandos mit vielen Argumenten können länger als eine Zeile sein. Für die bessere Übersicht werden diese Kommandos mehrzeilig abgedruckt und um zwei Zeichen eingerückt. Am Ende jeder Zeile steht der Backslash als Hinweis darauf, dass es in der nächsten Zeile weitergeht.

Rechtliches

Warennamen und Bezeichnungen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Es ist davon auszugehen, dass viele der Warennamen gleichzeitig eingetragene Warenzeichen oder als solche zu betrachten sind.

Bei der Zusammenstellung von Texten, Bildern und Daten wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Der Autor lehnt daher jede juristische Verantwortung oder Haftung ab. Für Verbesserungsvorschläge und Hinweise auf Fehler ist der Verfasser dankbar.

Einleitung

Cumulus Linux ist ein quelloffenes Netzwerk-Betriebssystem für Switches. Es basiert auf Debian GNU/Linux und vereint Techniken wie VXLAN, iptables, Kanalbündelung und Routing unter einer einheitlichen Kommandozeile. Cumulus Linux läuft auf physikalischer Hardware oder als virtuelle Maschine.

Jeder Ausrüster von Netzwerkkomponenten hat ein eigenes Betriebssystem im Angebot. Cumulus Networks verkauft Switches nur als Nebengeschäft. Die Grundidee ist: ein Linux-Betriebssystem für viele Hardwareplattformen anzubieten. Cumulus Linux ersetzt auf *anderen* Switches das Betriebssystem und kann dann loslegen.

In dieser Nische hat sich Cumulus Linux einen Namen gemacht. Dort punktet es in den Bereichen Funktionalität und Erweiterbarkeit. Cumulus Linux verbindet die Flexibilität von Linux mit den Anforderungen eines Datacenter-Switches bei moderater Budgetanforderung.

Cumulus Linux ist:

Unvollkommen. Und das ist positiv gemeint. Es gibt noch genug Raum zum Wachsen. Auch das Kommandointerface hat noch nicht alle Dienste unter seine Herrschaft gestellt. Das Feintuning von Funktionen erfordert meist einen Texteditor und den Zugriff auf die Konfigurationsdatei.

Open Source. Cumulus Networks setzt auf Linux als Betriebssystem und stellt viele seiner Eigenentwicklungen offen auf GitHub zur Schau [1]. Aber der Vorteil einer quelloffenen Lösung ist nicht immer ihr Preis. Denn

kostenlos ist Open-Source-Software nicht! Es fallen Lizenzgebühren an, und die Arbeitszeit der Netzwerk-Admins zum Einarbeiten in die Linux-Welt darf nicht unterschätzt werden.

Cumulus hält das Zusammenspiel mit dem Chipsatz (ASIC) der Switches unter dem Deckmantel des proprietären Codes. Und diese Geheimniskrämerei kommt vermutlich auf Wunsch der Hersteller, die ihre ASICs als Firmengeheimnis hüten.

Try before Buy. Wie bei Shareware-Programmen kann (und sollte) Cumulus Linux vor dem Einsatz getestet werden, bevor irgendwelche Investitionen in die Infrastruktur beginnen. Und wer freut sich über einen eingeschränkten Funktionsumfang, eine Evaluierungslizenz oder einen 30-Tage-Zeitraum?

In diesem Zusammenhang steht *Try* für Ausprobieren mit Beispielszenarien und *Buy* für den Einsatz in der eigenen Umgebung.

Hardware-frei. Cumulus Linux ist Software. Diese Software braucht eine Hardware. Die Antwort der Hardware-Frage liefert Cumulus mit einer Kompatibilitätsliste aus mehr als einhundert Geräten verschiedener Hersteller und Leistungsklassen.

In der Vergangenheit gab es viele limitierende Gründe, warum eine softwarebasierte Lösung für Netzwerkinfrastruktur nicht an die Leistung der physikalischen Geräte herankam. Der Hauptgrund war das suboptimale Zusammenspiel von Software und Treiber mit der darunterliegenden Hardware. Bei der immens großen Auswahl von Netzwerkkarten, Mainboards, Prozessoren und Memory ist es für eine Software schwierig auf jede Kombination der Komponenten optimal vorbereitet zu sein.

Heutzutage sind normale Server oder eingebettete Systeme überraschend performant, sodass auch eine nicht-

optimierte Software Bandbreiten jenseits von Gigabit durchbrechen kann.

Linux. Unter Cumulus läuft ein angepasstes Linux. Der Zugriff aufs Betriebssystem ist nicht gesperrt. Über das Konsolenmenü oder eine SSH-Verbindung und einem einfachen `sudo bash` liegt der Zugang offen.

Das bringt Möglichkeiten zum Anpassen, Verbessern und Nachinstallieren von Tools. Dagegen steht die Gefahr, dass die eigene Änderung ungewollte Instabilität mitbringt.

Best Of. Cumulus erfindet an vielen Stellen das Rad nicht neu und bedient sich für seine Features an den vertrauten Linux-Diensten, die nach Jahren der Entwicklung eine hohe Stabilität erreicht haben. Der Web-Server stammt von Nginx, der SSH-Server gehört zu OpenSSH und bei der Umsetzung der Firewallregeln helfen `netfilter` bzw. `iptables`.

Diebstahl? Keineswegs! Eher ein Nachweis, dass Open Source funktioniert. Solange Lizenzbedingungen eingehalten werden, darf Fremdsoftware beigemischt werden. Gerade im Security-Umfeld ist es höchst erwünscht, dass Anwendungsentwickler keine eigenen Implementierungen stricken, sondern sich an den freien und stabilen Bibliotheken bedienen.

White-Box-Switches

Ohne Hardware kann auch Cumulus Linux nichts ausrichten. Was im Serverumfeld gängige Praxis ist, erscheint in der Netzwelt skurril: Denn *White-Box-Switches*, oder *Bare-Metal-Switches*, sind Netzwerkgeräte *ohne* Betriebssystem. Damit ist nicht gemeint, dass das Betriebssystem vergessen wurde. Vielmehr hat der Kunde die freie Wahl und kann sich sein Wunschmodell so zusammenstellen, dass es in die

Infrastruktur von Rechenzentrum, Monitoring, Verwaltung und Automatisierung passt.

Diese Entkopplung von Hard- und Software hat auf beiden Seiten Vorteile. Egal welcher Hersteller von White-Box-Switches im Netzwerk mitspielt, das Betriebssystem sieht auf allen Boxen gleich aus. Das vereinfacht die Administration und den Lernaufwand, auch wenn die Architektur unterschiedlich ist.

Welches Betriebssystem darf es denn sein? Grundsätzlich läuft auf einem White-Box-Switch auch eine Distribution von Red Hat oder Ubuntu. Allerdings sind diese Anbieter nicht auf die Ausstattungen von Netzwerkschwitches vorbereitet. Cumulus Linux als Netzwerkbetriebssystem (*network operating system*, NOS) hat eine starke Ausrichtung auf Switchports, Buffer, SFPs, ASICs, CLI und die Überwachung von Temperatur und Lüfter.

Auf der Hardwareseite schonen White-Box-Switches das Budget. Diese Switches sind damit nicht billig, aber deutlich preisgünstiger als ein klassischer Switch von einem renommierten Netzwerkausrüster. Denn eine offene Preispolitik und das erklärte Ziel, die eigenen White-Box-Switches am Markt zu etablieren, bringen finanzielle Vorteile für die Käufer.

Geschichte

Die Historie von Cumulus Linux ist noch relativ kurz: Es beginnt 2010 in den USA mit der Initiative von ehemaligen Mitarbeitern von Cisco und VMware. Ihr gegründetes Start-up-Unternehmen will ein offenes Betriebssystem für Netzwerkschwitches entwickeln, das auf Datacenter-Switches von namhaften Herstellern operiert, erweiterbar ist und eine einheitliche Verwaltung bietet.

Die Entwicklung zieht sich hin, aber im Juni 2013 veröffentlicht Cumulus ihr Konzept des offenen Switches und

beginnt mit der ersten Version 1.5. Die Branche sieht es als Angriff auf teure und proprietäre Hersteller. Vereinzelt spricht man sogar von Cumulus Linux als *Cisco Killer*.

In den folgenden Jahren beschäftigt sich Cumulus mit Entwickeln und Klinkenputzen. Denn gute Software braucht Partner und Anpassungen an die jeweilige Hardware. So kommt es 2014 zu einer Partnerschaft mit *Dell*, worauf Cumulus Linux für die S-Serie optimiert angeboten wird. 2015 folgt *Hewlett Packard* und die Altoline-Switches werden Cumulus-ready. Im Folgejahr stößt Mellanox mit den Spectrum-Switches dazu. Zuletzt einigt man sich mit Lenovo auf eine Zusammenarbeit mit den ThinkSystem-RackSwitches.

Parallel dazu wird Cumulus Linux funktionsreicher und stabiler. Im August 2015 macht Cumulus einen freundlichen Schritt und bietet ihr Betriebssystem für Laborumgebungen kostenlos an. Das als *Cumulus VX* angebotene Linux ist eine virtuelle Maschine, mit der einzelne Funktionen und ganze Netzdesigns ausprobiert werden können, ohne dass die produktive Umgebung dafür erhalten muss.

Zur Begriffsklärung: *Cumulus Networks* ist der Firmenname. Die Linuxdistribution für Switching auf physikalischen Geräten heißt *Cumulus Linux*. Als *Cumulus VX* bietet der Hersteller seine Software als virtuelle Appliance an, die ohne Lizenzkosten (und Support) auf den bekannten Hypervisoren ausprobiert werden kann. Die Bezeichnung *Cumulus Express* ist ein Bündel aus *Cumulus Linux*-Betriebssystem, Switch-Hardware und passender Lizenz. Cumulus Linux hat zum Entwicklungsbeginn bei Versionsnummer 1.5 gestartet und ist bisher (2019) bei Version 3.7.6 angekommen.

Für eine Webrecherche ist das Schlagwort „cumulus-linux“ am aussagestärksten.

Teil I

Für Einsteiger

Kapitel 1

Das Labornetzwerk

Vor dem Einstieg in den Umgang mit Cumulus Linux steht der Aufbau des Labornetzwerks, denn ein einzelner Switch ohne umgebendes Netzwerk ist wenig beeindruckend. Für den praxisnahen Einstieg erwacht Cumulus Linux in einem konstruierten Labornetz zum Leben. In dieser Umgebung kann Cumulus Kapitel für Kapitel mit seinen Fähigkeiten glänzen.

Alle Themen der Kapitel haben einen praktischen Hintergrund. Theoretische Grundlagen werden nur am Anfang eines Kapitels angesprochen, um Verständnis aufzubauen oder angestaubtes Wissen aufzufrischen. Die Beispiele und Übungen sind zum Nachspielen konzipiert.

Die Passagen basieren alle auf demselben Netzaufbau. Es stellt ein kleines Rechenzentrumsnetz mit redundanten Verbindungen dar. Je nach Komplexität eines Themas reicht ein Teil des Labornetzwerks aus, um die Kernaussage zu beschreiben.

Wenn ein Abschnitt einen gesonderten Aufbau benötigt oder ein weiteres Gerät untersucht werden soll, gibt es am Anfang der Lektion einen entsprechenden Hinweis mit Erklärung.

Ressourcen

Der stets unveränderte Aufbau des Labornetzes hat den charmanten Vorteil, dass zwischen den Kapiteln nicht umgebaut werden muss. Kein Umverkabeln der Geräte oder Umkonfigurieren der virtuellen Umgebung. Das spart Zeit und verhindert Fehler. Und nach ein paar Kapiteln wird das Labornetz zum vertrauten Begleiter, denn die Namen der Switches, Server, Netzschnittstellen und IP-Adressen bleiben unverändert.

Das vollständige Labornetz ist als Netzdiagramm in [Abbildung 1.1](#) dargestellt. Es ist als Grundlage für die nachfolgenden Kapitel konzipiert und orientiert sich an der Spine-Leaf-Architektur (vgl. [Kap. 14](#)). In den folgenden Kapiteln werden meist nur Teile dieses Netzwerks zur Untersuchung benutzt.

Da ein händischer Eingriff nach dem ersten Aufbau nicht mehr notwendig ist, kann das Lab auch „aus der Ferne“ betrieben werden – Remotezugriff vorausgesetzt.

Die erforderliche Hardware ist stets abhängig vom geplanten Durchsatz. Für die Laborgeräte eignet sich jedes Gerät von der Kompatibilitätsliste [\[2\]](#). Für virtuelle Umgebungen stellt Cumulus eine passende virtuelle Appliance zur Verfügung.

Manche Kapitel arbeiten isoliert, andere benötigen Internetzugang. Der Zugang zum Internet läuft stets über das Managementinterface *eth0*. Hier reicht ein Uplink zum DSL-Router, aber grundsätzlich ist alles möglich, was letztendlich ins Internet führt.

Virtualisierung

Alle Geräte im Lab können vollständig virtualisiert werden. Jeder Switch im Labornetz ist dann eine eigene virtuelle Maschine (VM) mit virtuellen Netzkabeln zu den

benachbarten VMs. Das Verbindungsnetz zwischen zwei VMs ist ein *LAN Segment* (bei VMware) oder *internes Netzwerk* (bei VirtualBox). Eine physikalische Netzwerkkarte im Hostsystem ist nötig, wenn mit echter Hardware gemischt wird.

Welches Interface in welchem virtuellen Netz Zuhause ist zeigt [Tabelle 1.1](#) auf Seite →.

Technisch nicht erforderlich, aber hilfreich zum Auswerten: Die Netzwerkkarten der VMs verwenden vordefinierte MAC-Adressen. Damit sind alle Geräte in den Kommandoausgaben eindeutig erkennbar und mit den Beispielen im Buch vergleichbar.

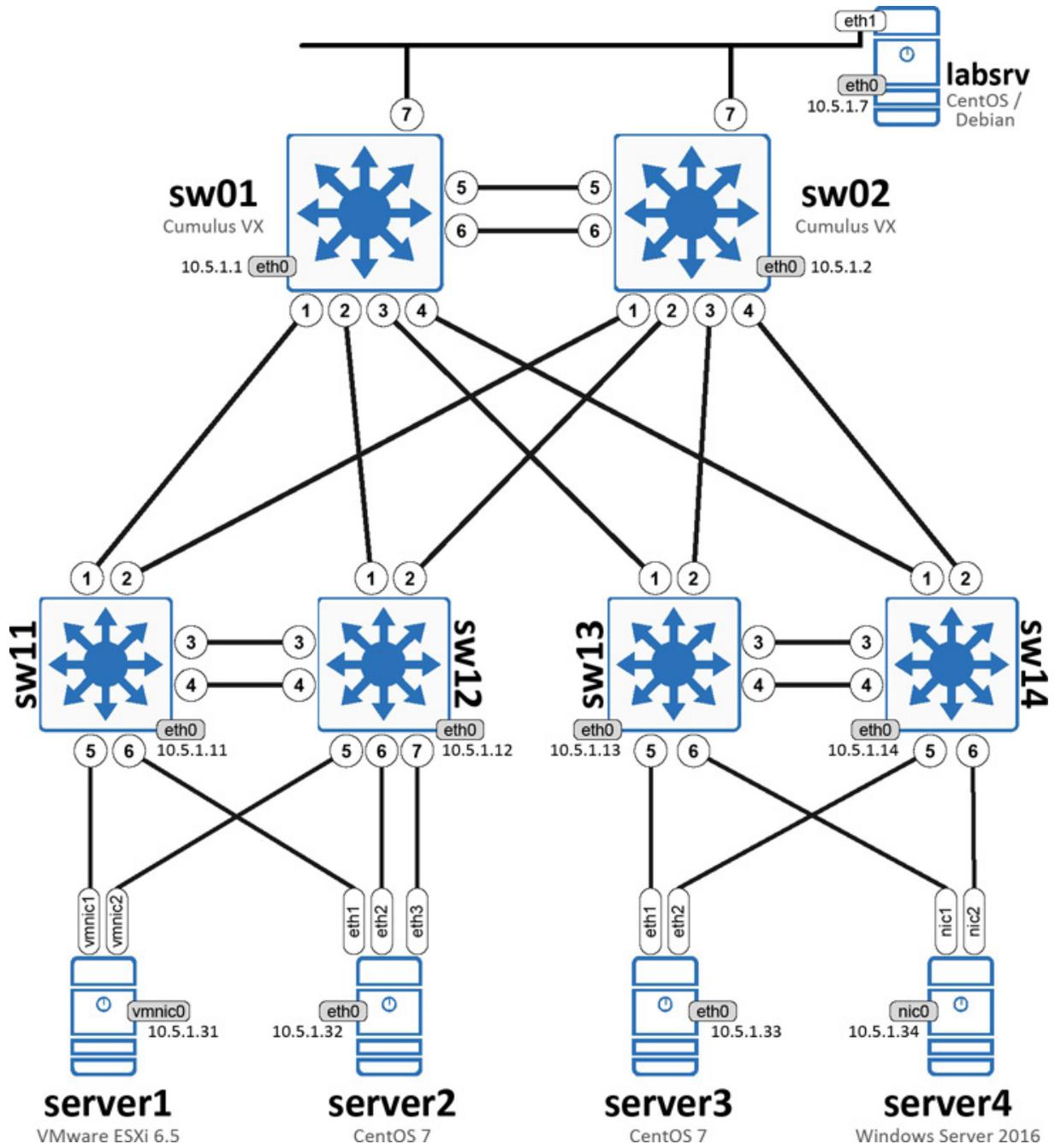


Abbildung 1.1: Das Labornetzwerk als Vorlage für alle Kapitel

Switch	Interface	Funktion/Netz	IPv4	IPv6
sw01	eth0 swp1-4 swp5,6 swp7	Management Spine-Leaf Spine-Spine VMnet4	10.5.1.1 10.4.1.1	fd00:5::1 fd00:4::1
sw02	eth0 swp1-4 swp5,6 swp7	Management Spine-Leaf Spine-Spine VMnet4	10.5.1.2 10.4.1.2	fd00:5::2 fd00:4::2
sw11	eth0 swp1,2 swp3,4 swp5,6	Management Leaf-Spine Leaf-Leaf Leaf-Server	10.5.1.11	fd00:5::11
sw12	eth0 swp1,2 swp3,4 swp5-7	Management Leaf-Spine Leaf-Leaf Leaf-Server	10.5.1.12	fd00:5::12
sw13	eth0 swp1,2 swp3,4 swp5,6	Management Leaf-Spine Leaf-Leaf Leaf-Server	10.5.1.13	fd00:5::13
sw14	eth0 swp1,2 swp3,4 swp5,6	Management Leaf-Spine Leaf-Leaf Leaf-Server	10.5.1.14	fd00:5::14
labsrv	eth0 eth1	Management VMnet4	10.5.1.7 10.4.1.7	fd00:5::7 fd00:4::7
server1	vmnic0 vmnic1,2	Management Server-Leaf	10.5.1.31	
server2	eth0 eth1-3	Management Server-Leaf	10.5.1.32	
server3	eth0 eth1,2	Management Server-Leaf	10.5.1.33	
server4	Ethernet0 Ethernet1,2	Management Server-Leaf	10.5.1.34	

Tabelle 1.1: Alle Switches mit Netzadaptern, Funktion und Managementadresse

Getestet und geprüft sind die Labs mit VMware Workstation 14, VMware ESXi 6.5 und VirtualBox 5.1.

Hardware

Cumulus Linux läuft auf physikalischen Switches mit ARM-, PowerPC- oder x86_64-Prozessor. Bei der Auswahl von passender Hardware lohnt sich ein Blick in die Kompatibilitätstmatrix von Cumulus Linux [2]. Für die Laborumgebung sind Hersteller und Modell zweitrangig, da das Beispielnetz Verständnis bieten soll und nicht Höchstleistung.

Die folgenden Kapitel verwenden Switches vom Hersteller Edgecore.

Netze

Die Netze zwischen den Switches basieren auf Ethernet. Jede Verbindung zwischen zwei Geräten besteht aus einem Kabel ohne weitere Teilnehmer. Das Übertragungsmedium und die Bandbreite spielen keine Rolle. Kupferkabel sind ebenso willkommen wie Glasfaserkabel.

Im Beispielnetz sind alle Netzverbindungen Kupferkabel mit einer Übertragungsrage von einem Gbit/s.

Switches

Die Cumulus-Switches verwenden die aktuelle Version 3.7.6. Wenn andere Versionen oder zusätzliche Switches mitspielen, wird das entsprechende Gerät ersetzt oder das Lab ergänzt.

Hinweis

Die beste Version eines Betriebssystems ist nicht immer die Aktuellste. Die Release Notes und ein Softwaretest in

einer realistischer Umgebung liefern Entscheidungsgrundlagen für eine stabile Version.
--

Jeder Switch hat einen separaten Netzanschluss für den Managementzugriff. Darüber erreicht der SSH-Client sein Ziel und kann Konfigurationsänderungen unabhängig von der Topologie umsetzen.

Die Labor-Switches sind durchnummeriert. Diese Geräte-Nummer findet sich in den IPv4-, IPv6- und MAC-Adressen wieder. Damit sind Adressen in einer Kommandoausgabe leichter dem passenden Gerät zuzuordnen. Die Nummer des Switchports ist stets am Gerätesymbol angeschlagen.

Adressierung

Die Subnetze für Server und Transitbereiche bauen auf private IPv4-Adressen bzw. Unique-Local IPv6-Adressen. Jeder Leaf-Switch verbindet mehrere Server, die meist nur zum Prüfen von Features oder zum Erzeugen von Datenverkehr benutzt werden. Mehr als ping, traceroute, netstat oder ein Webbrowser wird nicht gefordert.

Die Wahl des Betriebssystems der Server ist für die Szenarien nebensächlich; im Demo-Lab finden aus Popularitätsgründen VMware ESXi, CentOS Linux und Windows Server Verwendung.

Wenn das Labornetz zwischen internen und externen Netzbereichen unterscheidet, dann bedienen sich die externen Geräte aus den Adressblöcken für Dokumentation (RFC 5737): 192.0.2.0/24 und 198.51.100.0/24.

Die IPv6-Adressen stammen ebenfalls aus unterschiedlichen Bereichen, um eine Unterscheidung optisch zu vereinfachen: fd00::/16 für die internen LANs und 2001:db8::/32 für die äußeren Bereiche. Diese