

# FreeBSD auf dem Desktop

## Einstiegshilfe zur Installation und Konfiguration mit zahlreichen Beispielen

### Vorwort

Erst seit Mitte 2015 befasse ich mich mit FreeBSD<sup>1</sup>. Davor habe ich lange Zeit GNU/Linux<sup>2</sup> als Haupt-System genutzt, mit S.u.S.E 5.3<sup>3</sup> habe ich angefangen, dann bin ich wenige Jahre später bei Debian gelandet, und dies sehr lange unter Nutzung verschiedener Zweige als auch Derivate. Gelegentliche Ausflüge zu PCLOS, Mageia, Manjaro und Arch hat es auch gegeben.

Da ich kein großer Freund des neuen Init-Systems **systemd**<sup>4</sup> bin, das sich mittlerweile bei vielen GNU/Linux-Distributionen quasi zum Standard etabliert hat, nutze ich als primäres GNU/Linux System seit einigen Monaten MX-Linux<sup>5</sup>.

Letztes Jahr habe ich dann doch aus Neugier damit begonnen, mir FreeBSD näher anzuschauen (ein Ausflug zu PC-BSD<sup>6</sup> davor war nur von kurzer Dauer, dieses System ist mir etwas zu überladen, und wenn man zu FreeBSD lernen möchte, eignet sich wohl das Original besser, als ein Derivat)<sup>7</sup>. Nach einiger Lektüre und einigen Installationsversuchen, die auch zu laufenden benutzbaren Systemen geführt haben, habe ich mich dann beim englischsprachigen Forum<sup>8</sup> angemeldet, um eine Kommunikationsplattform zu diesem unixartigen System zu haben. Neben Themen mit Fragen hat es dann auch einige Auseinandersetzungen gegeben. Folgenden Fehler habe ich dabei gemacht:

---

1 Siehe <https://www.freebsd.org/de/>

2 Siehe <https://www.gnu.org/gnu/linux-and-gnu.de.html>

3 Anm.: Diese Distribution aus Nürnberg kam mit mehreren CD-Roms und einem dicken Benutzerhandbuch. Meines Erachtens bot sie einen sehr schönen Einstieg zu diesem unixartigen Betriebssystem.

4 Anm.: Zu diesem Init-System - siehe auch <https://de.wikipedia.org/wiki/Systemd> - gibt es diverse Auseinandersetzungen, neben sachlichen Pro- und Kontra-Meinungen aus Entwickler- wie auch Nutzersicht gibt es ausgesprochen unproduktive Schlammschlachten. Ich glaube einerseits durchaus, dass **systemd** interessante Features hat, die Dinge erleichtern können. Allerdings halte ich es andererseits für eine Fehlentwicklung, wenn man für einfach zu handhabbare Dinge ein hochkomplexes Werkzeug benutzt. Davon abgesehen finde ich es bedenklich, wenn sich allmählich in größeren Teilen der GNU/Linux-Szene eine Auffassung durchsetzt, dass Kritik an diesem Init-System eigentlich nur von Volltrotzeln kommen kann. Da ich in Sachen **systemd** trotz meiner Vorbehalte gerne auf dem Laufenden bleiben möchte, habe ich noch ein Antergos laufen, das ist weitgehend ein Vanilla-Arch plus einigen Zusätzen, das sich dank des Installers Calamares - siehe <https://calamares.io/> - rasch installieren lässt.

5 Siehe <http://www.mepiscommunity.org/mx>

6 Anm.: PC-BSD basiert auf FreeBSD, die Entwickler haben den Anspruch, ein installationsfreundliches BSD für Desktop-Systeme und Server-Systeme zu schaffen, neben FreeBSD kommen Eigenentwicklungen hinzu, siehe dazu auch <https://web.pcbbsd.org/>. Eine meines Erachtens spannende Eigenentwicklung, der ein eigenes Kapitel dieses Buches gewidmet sein wird, ist der unter einer BSD-Lizenz stehende Desktop Lumina, siehe dazu <https://lumina-desktop.org>

7 Anm.: Ich möchte PC-BSD keinesfalls schlecht machen, wenn das schlanke Desktop-Environment Lumina zum Einsatz kommt, ist das Argument mit der Überladenheit wohl nicht mehr zutreffend.

8 Siehe <https://forums.freebsd.org/>

Ich bin mit der Erwartung an FreeBSD heran gegangen, dass es zum einen ähnlich zu funktionieren hat wie GNU/Linux und zum anderen ähnlich dicht dokumentiert zu sein hat, wie GNU/Linux.<sup>9</sup> Das sehe ich mittlerweile anders, davon abgesehen hat es aber bei den Diskussionen einen Aspekt gegeben, den sogar der Foren-Admin Warren Block<sup>10</sup> aufgegriffen und verteidigt hat:

Dokumentationen zu FreeBSD können auch so gemacht sein, dass sie von Anfängerinnen und Anfängern ohne große Vorkenntnisse abzarbeiten sind und dass dabei dann funktionierende Systeme heraus kommen.

Zwei sehr hilfreiche Sachen habe ich zu jener Zeit allerdings noch nicht gekannt, und zwar eine Installationsanleitung zu FreeBSD auf dem Desktop von Nicole<sup>11</sup> und die Seiten zur FreeBSD-Installation von Martin<sup>12</sup>

Abgesehen davon ist es ja schön, zu fordern, dass etwas **so gemacht sein kann**, aber noch etwas anderes, nicht nur zu fordern, sondern es **dann auch zu machen**. Ich habe meine Installationen von FreeBSD bisher immer stichpunktartig auf Papier notiert, heraus gekommen ist eine lose Sammlung von dahin gekritzelteten Seiten, ein Teil dieser Sammlung ist dann von einem unserer Kater genüsslich zerfleddert worden. Der Kater lebt noch, ich bin dem Tierchen sogar ein wenig dankbar, denn danach habe ich überlegt, ob ich nicht meine Installationen mit dem PC protokollieren und dann auch als Dokumentation zur Verfügung stellen soll. Zunächst habe ich geplant, zur FreeBSD-Installation auf meinem Desktop-Rechner eine Dokumentation zu schreiben, inklusive einem experimentellen Kapitel zu plasma5.<sup>13</sup> Diese Dokumentation sollte schon über eine reine Beschreibung hinaus gehen, aber nicht allzu umfangreich werden. Eine erste Rohfassung habe ich dann im deutschsprachigen BSD-Forum zur Diskussion gestellt,<sup>14</sup> allen Beteiligten bin ich dankbar für die freundlichen Rückmeldungen, sachlichen Ergänzungen und kritischen Anmerkungen.

Dieses Feedback hat Konsequenzen, und zwar den fast vollständigen Umbau der Dokumentation. Im Vordergrund steht nach wie vor mein Bemühen, einsteigerfreundlich über FreeBSD auf dem Desktop zu schreiben, die eigenen Installationen spielen auch weiterhin eine Rolle, sie stehen aber nicht mehr zu sehr im Mittelpunkt, sondern dienen eher als anschauliche Beispiele. Wichtig ist mir dabei folgendes (das habe ich auch schon während einer Diskussion auf BSD-Foren geäußert):

---

9 Anm.: Ich habe Beispiele gebracht, bei denen Schlagworte wie „xyz + Arch“ in Suchmaschinen sofort brauchbare Treffer liefern, man bei FreeBSD hingegen manchmal zunächst im Regen steht. Nun hat Arch ein spitzenmäßiges Wiki, es wird oftmals auch von Leuten anderer GNU/Linux-Distributionen bei kniffligen Angelegenheiten genutzt.

10 Anm.: Warren Block hat eine Seite mit einer Sammlung von Howtos - siehe <http://www.wonkity.com/~wblock/docs/> - ich kann sie nur empfehlen und werde darauf noch zurück kommen.

11 Siehe <https://cooltrainer.org/a-freebsd-desktop-howto/>

12 Siehe <http://denkrobat.de/doku.php>

13 Anm.: Unter FreeBSD ist es dank des inoffiziellen area51 Projektes möglich, sehr aktuelle Versionen von plasma5 samt KDE-Applikationen zu bekommen, siehe dazu <https://freebsd.kde.org/news.php> und <http://area51.pcbsd.org/branches/plasma5/README>

14 Siehe <https://www.bsdforen.de/threads/freebsd-auf-dem-desktop-von-einem-einsteiger-f%C3%BCr-einsteiger-innen.32670/>

*„[...]Was ich da schreibe, möchte ich auch ordentlich verstanden haben und was ich schreibe, soll auch für andere nachvollziehbar sein und dann bei den anderen auch funktionieren. Ich finde es sehr abschreckend, wenn Leute, die mit unixoiden Systemen beginnen, auf Doku stoßen, die schwierig nachzuvollziehen ist und die viel zu viel voraus setzt. Daher meide ich auch weitgehend während meiner Erklärungen zu einer Sache Querverweise, ich möchte nicht, dass man zu sogenanntem Howto-Hopping gezwungen ist, um bei der Installation weiter zu kommen (Doku a verweist bei einer Sache auf Handbuch x und dort muss dann bei Manpage zu z nachgeschaut werden, so was mag für Profis keine Schwierigkeit sein, aber für Leute, die anfangen, kann das frustrieren), sondern anhand der Doku soll jede / jeder schrittweise ein System aufsetzen können und auch dabei begreifen, was sie / er jeweils tut (Hinweise auf Handbuch, Manpages oder Howtos sind bei mir immer als Ergänzung und Wissenserweiterung zu verstehen).[...]“<sup>15</sup>*

Eine anfangs geplante Änderung habe ich vorerst nicht umgesetzt und die an dieser eigentlich geplanten Änderung Beteiligten mögen mir das nachsehen: Ich werde nun doch vorerst mit Hilfe von LibreOffice diese Dokumentation schreiben. LyX,<sup>16</sup> LaTeX<sup>17</sup> und ähnliche professionelle Software ist eindeutig mächtiger und wohl auch besser für mein Vorhaben geeignet. Allerdings bedarf es auch einer gewissen Einarbeitungszeit. Und ich habe, was den Inhalt der Dokumentation angeht, mich noch in so viele Dinge wenigstens grundlegend einzuarbeiten,<sup>18</sup> dass es mir einfach zu viel wird, daneben mich auch noch mit LaTeX vertraut zu machen. Doch „vorerst mit LibreOffice“ heißt ja nicht, dass ich auf LyX verzichten möchte, sollte ich etwas mehr Zeit haben – momentan ist das allerdings nicht absehbar – dann werde ich schon fertige Teile der Dokumentation nach LyX übertragen.

Weiterhin basieren die hier nun verwendeten Installationen von FreeBSD auf der bald veröffentlichten Version FreeBSD 11 und nicht, wie zu Beginn geplant, auf FreeBSD 10.3.<sup>19</sup>

---

15 Siehe <https://www.bsdforen.de/threads/freebsd-auf-dem-desktop-von-einem-einsteiger-f%C3%BCr-einsteiger-innen.32670/page-2#post-282001>

16 Siehe <https://www.lyx.org/>

17 Siehe <https://www.latex-project.org/>

18 Anm.: Zu erwähnen ist hier etwa das Dateisystem ZFS, der Paketbau mit **poudriere** und ähnliches.

19 Anm.: Dieses Vorwort habe ich Mitte Juli 2016 geschrieben, Anfang September 2016 wird FreeBSD 11 veröffentlicht.

# Einleitung

Das Buch soll ein ausführlicher Leitfaden zur Installation von FreeBSD auf Desktop-Rechnern oder Notebooks sein, angereichert mit zahlreichen Beispielen,<sup>20</sup> und richtet sich an Anfängerinnen und Anfänger.<sup>21</sup>

Bevor ich richtig loslege, sei vorweg gestellt, was das Buch nicht sein soll:

Dieses Buch habe ich nicht als Konkurrenz zum FreeBSD-Handbuch konzipiert, es bietet vermutlich für FreeBSD-Profies kaum neue Erkenntnisse. Da ich selbst noch Anfänger bin – wenn auch ein schon etwas fortgeschrittener – bitte ich um Nachsicht und auch Korrektur, falls hier einige Sachen nicht optimal oder gar fehlerhaft umgesetzt sind, und es in jenen Fällen deutlich bessere Möglichkeiten einer Konfiguration gibt.

Der Aufbau ist sukzessiv, es soll ein umfangreicher sogenannter roter Faden zu FreeBSD auf dem Desktop entstehen. Manche Kapitel beginnen mit theoretischen Abschnitten, wer es eher eilig hat, kann natürlich einiges überspringen, ich empfehle dies aber nicht, weil ich mich bemühe, in diesen Abschnitten Sachen auch so zu erläutern, dass sie in den dazu passenden praktischen Teilen als hilfreiches Hintergrundwissen fungieren. Einige Kapitel sind vielleicht für manche Leserinnen und Leser nicht weiter interessant, wer sich zum Beispiel für UFS2 als Dateisystem entschieden hat, mag sich eventuell nicht weiter mit dem Kapitel zu ZFS befassen.

Ich fange mit einem **Kapitel zur Basiseinrichtung** des Systems an:

- Neben Bezugsquellen von FreeBSD geht es um eine Vorbereitung von Installationsmedien.
- Weiter geht es dann mit dem ersten Start von FreeBSD über ein Installationsmedium und dem Prüfen der eigenen Hardware mit verschiedenen Werkzeugen, die FreeBSD schon auf dem Installationsmedium mitbringt.
- Danach stelle ich den Standardinstaller **bsdinstall**<sup>22</sup> vor, ich gehe dann eine Installation des Basissystems anhand meines Desktop-Rechners durch. Ein größeres Unterkapitel widme ich dabei den zu verwendenden Dateisystemen und der oft zu lesenden Frage, wie denn partitioniert werden soll. Dabei spiele ich verschiedene Partitionierungen / Einrichtungen der Festplatte durch.<sup>23</sup> Eine Erläuterung der weiteren Schritte einer Basis-Installation bis zum ersten Start des neu installierten FreeBSD-Systems rundet das erste Kapitel ab.<sup>24</sup>

---

20 Anm.: Die Beispiele beziehen sich auf unsere PCs, Notebooks und unser Netzwerk, ein Übertragen auf andere PC Situationen und Nutzerbedürfnisse sollte aber nicht allzu schwierig sein.

21 Anm.: Wenn erforderlich, gebe ich weitere Quellen als Referenzen an, z.B. das FreeBSD Handbuch (englische Fassung), Manuals z.B. von Warren Block, von @Rakor, oder verweise auf Diskussionen in den <https://www.bsdforen.de> oder <https://forums.freebsd.org/>

22 Manpage <https://www.freebsd.org/cgi/man.cgi?bsdinstall>

23 Anm.: Bei der Verwendung des Dateisystems zfs kann man nicht mehr im klassischen Sinn von Partitionieren sprechen. Mehr zu Dateisystemen im weiteren Verlauf.

24 Anm.: Der FreeBSD-Installer ist weitgehend selbsterklärend und meines Erachtens so gut gemacht, dass ich dazu nicht viel schreiben werde.

Das **zweite große Kapitel umfasst dann den weiteren Aufbau des Systems** nach der Basisinstallation:

- Ich beginne mit Erläuterungen zu wichtigen Konfigurationsdateien und den Möglichkeiten der Edition dieser Dateien, die die Basisinstallation schon mitbringt.
- Dann ist die Einrichtung neuer Gruppen und Nutzerkonten dran, samt der Befugnisse, die man neuen Nutzern gestattet.
- Umfangreich wird das Unterkapitel zum Paketmanagement und welche Möglichkeiten man unter FreeBSD hat, neue Software hinzu zu installieren.<sup>25</sup>

Im **dritten großen Kapitel widme ich mich dann dem Desktop**.

- Nach Vorbemerkungen zur Unterstützung verschiedener Grafikkarten fange ich mit der Installation und Konfiguration des Xservers an.
- Weiter geht es mit der Soundkarte. Die bisher von mir verwendeten Soundchips ließen sich problemlos mit FreeBSD zum Erklingen bringen.
- Das folgende sehr umfangreiche Kapitel widmet sich der jeweiligen Installation und Konfiguration einzelner Desktop-Umgebungen.<sup>26</sup> Ausgewählt habe ich KDE4, Xfce4, Lxde und Lumina. Eventuell kommt noch Gnome hinzu, allerdings bin ich mit Gnome nicht sehr vertraut und habe es bisher auch unter GNU/Linux eher gemieden.<sup>27</sup>
- Weiter geht es mit typischer Software auf einem Desktop-Rechner wie Mailprogrammen, Browsern, Office-Suiten, ich gehe dabei auch auf Software rund um Audio, Video und Bilder ein.
- Zu einem Desktop-System gehört nicht selten auch die Verwendung eines oder sogar mehrerer lokal angeschlossener Drucker. Daher befaße ich mich auch mit der Einrichtung von Cups und der Druckerkonfiguration auf FreeBSD.<sup>28</sup>

Das **vierte Kapitel behandelt fortgeschrittene Themen**.

- Zunächst geht es um die Integration eines FreeBSD-Desktop-Rechners in ein bestehendes Netzwerk, ich thematisiere ausführlicher ssh-Zugriff auf andere Rechner, gefolgt von einem Unterkapitel zu Remote-Printing, also der Anbindung an einen Print-Server.

---

25 Anm.: In diesem Unterkapitel behandle ich pkg als zentrales Installationswerkzeug von FreeBSD, gefolgt von der Nutzung nur vorkompilierter Pakete für eine Installation (packages-only) oder das Einbeziehen von Ports (builds+packages) oder dem Komplettbau aus Ports (builds-only). Schon hier sei erwähnt, dass für die eigentliche Installation immer pkg zuständig ist, daher hat man bei allen Installationsvarianten immer eine pkg-Datenbank, in der alle installierten Pakete vermerkt sind und installiert daher nichts am Paketsystem vorbei.

26 Anm.: Ausführlich eingehen werde ich auf KDE4, Xfce4, Lxde und Lumina.

27 Anm.: Damit meiner ich nicht, dass ich es mir gar nicht erst angeschaut habe, nach kurzer Zeit musste bisher Gnome2 / Gnome3 bei mir meistens wieder KDE4 oder manchmal Xfce4 weichen, weil ich mit diesem Desktop-Environment einfach nicht richtig warm werde.

28 Anm.: Wer unter GNU/Linux mit Cups ordentlich eingerichtete Drucker laufen hatte, muss unter FreeBSD mit Cups keine Abstriche machen. Auf andere Möglichkeiten, Drucker einzurichten, gehe ich nur am Rande ein.

- Ich dokumentiere dann die Migration unseres Daten-NAS von Debian GNU/Linux nach FreeBSD, dafür richte ich wieder ein Root-on-ZFS System ein, allerdings diesmal mit Redundanz, weil ich den Pool spiegeln möchte.
- Breit angelegt sein wird das Unterkapitel zur Einrichtung und Pflege eines lokalen Repositoriums. Zunächst werde ich zeigen, wie man mit **poudriere**<sup>29</sup> einzelne Pakete oder Paketserien sauber bauen, lokale Repos mit diesen Paketen befüllen und Client-Rechnern in einem Netzwerk zur Verfügung stellen kann. Es folgt als Beispiel der Bau sämtlicher Plasma5-Pakete aus dem noch inoffiziellen Area51-Port.<sup>30</sup>

---

29 Siehe <https://www.freebsd.org/doc/handbook/ports-poudriere.html> und das Wiki dazu <https://wiki.bsdforen.de/kategorie:howto:poudriere>

30 Siehe <https://freebsd.kde.org/> und <http://area51.pcbsd.org/branches/plasma5/README>

# Inhaltsverzeichnis

1 Die Basisinstallation von FreeBSD.....	8
1.1 Bezugsquellen.....	8
1.2 Der erste Start – FreeBSD und Hardware-Unterstützung.....	10
1.3 Und nun auf eine Festplatte damit: Zur Installation.....	11
1.3.1 Das Tastatur-Layout.....	11
1.3.2 Wie soll der Rechner heißen?.....	11
1.3.3 Die Partitionierung.....	11
1.3.3.1 Dateisysteme: Der Klassiker UFS2.....	12
1.3.3.1.1 UFS2 für Bequeme: Automatische Einrichtung durch den Installer....	12
1.3.3.1.2 Manuelle Partitionierung einer Festplatte mit der Shell.....	12
1.3.3.2 ZFS: Ein innovatives Dateisystem.....	14
1.3.3.2.1 ZFS für Bequeme: Automatische Einrichtung durch den Installer....	14
1.3.3.2.2 Root-on-ZFS: Manuelle Einrichtung.....	14
1.3.3.2.2.1 Erzeugen des Pools und der Data-Sets.....	14
1.3.3.2.2.2 Letzte Schritte bei manueller Root-on-ZFS Einrichtung.....	16
1.3.4 Die Paßwortvergabe für den Administrator.....	16
1.3.5 Auswahl einiger zu startender Dienste.....	16
2 Die Konfiguration von FreeBSD.....	17
3 FreeBSD als Desktop-System.....	17
4 Fortgeschrittene Themen.....	17
4.1 FreeBSD im eigenen Netzwerk.....	17
4.2 Migration eines Dateiservers von GNU/Linux nach FreeBSD.....	17
4.3 Anlegen und pflegen eigener Repositorien.....	17
4.3.1 Paketbau mit poudriere.....	17

# 1 Die Basisinstallation von FreeBSD

Die diesem Buch zugrunde liegende Referenz-Hardware (verschiedene Rechner und Notebooks von meiner Frau und mir mit Intel- und Nvidia-Grafik) wird ordentlich von FreeBSD unterstützt, zu Grafikkarten von Ati / AMD kann ich mangels Hardware keine Angaben machen. Was mir fremde Hardware angeht, sei folgendes angemerkt:

Da diese Dokumentation sehr auf meine Erfahrungen mit eigenen Installationen ausgerichtet ist, kann ich natürlich auch nur über solche Hardware ausführlich schreiben, die ich selbst habe. Mit Vermutungen über mir fremde Hardware, zur der ich nur etwas gelesen habe, möchte ich mich eher zurückhalten, es steht leider auch einiges im Internet dazu, was entweder veraltet ist, ungeprüft abgeschrieben oder nur gemutmaßt wurde. Es ist meines Erachtens ausgesprochen ärgerlich, Zeit mit Hinweisen und Anleitungen zu verbringen, die dann nicht greifen und ich möchte mich mit meiner Dokumentation nicht in diese gerade für Anfängerinnen und Anfänger frustrierende Kategorie von Anleitungen einreihen.

Wenn mir allerdings Leute von ihrer mir fremden Hardware detailliert berichten, dass sie ordentlich unter FreeBSD zum Laufen zu bekommen ist, nehme ich diese Hinweise gerne auf, selbstverständlich mit Nennung der Quelle.<sup>31</sup>

Bevor ihr nun aber herum rätselt, ob eure eigene Hardware für eine Verwendung von FreeBSD geeignet ist, könnt ihr das hervorragend testen. Daher fange ich nun zunächst mit den Bezugsquellen und dem Vorbereiten von Installationsmedien an.

## 1.1 Bezugsquellen

**Info Werkzeuge:** Falls ihr bisher nur mit Windows gearbeitet habt, macht euch bitte über ein geeignetes Werkzeug zum Schreiben von USB-Sticks mit Images schlau, denn da ich kein Betriebssystem von Microsoft nutze, kann ich hier mangels Erfahrungen und Testmöglichkeiten keine praktischen Tipps dazu geben.<sup>32</sup> Unter GNU/Linux oder einem anderen unix-artigen System benötigt ihr zum Erstellen eines Installationsmediums auf USB-Stick das Programm **dd**<sup>33</sup>

Damit nicht unnötig Rohlinge vergeudet werden – vielleicht wollt ihr FreeBSD nach einigen Tests ja gar nicht auf euren Rechnern installieren – und damit während der Tests eure Nerven geschont werden (heulende DVD-Laufwerke sind verzichtbar), empfehle ich die Verwendung eines USB-Sticks.<sup>34</sup> Passende Images sind unter folgender URL zu finden:

<http://www1.de.freebsd.org/freebsd/releases/ISO-IMAGES/11.0/>

31 Anm.: Es ist meines Erachtens selbstverständlich und ich bemühe mich daher auch darum, beim Verwenden anderer Quellen wie Dokumentationen, Hinweise in Form von Forenbeiträgen u.s.f. auf diese auch per Fußnote hinzuweisen.

32 Anm.: Hier könnte ich auch nur eine Suchmaschine bemühen.

33 Siehe <http://man7.org/linux/man-pages/man1/dd.1.html>

34 Anm.: Falls ihr nicht von USB-Medien booten könnt, habt ihr auch die Möglichkeit, optische Medien wie CDs oder DVDs zur Installation zu verwenden. Wichtig beim Brennen der Images auf die Medien ist die Auswahl „Brennen eines ISO-Abbildes“ oder ähnliches in eurem Brennprogramm. Kopiert ihr die Images bloß auf CD oder DVD, erzeugt ihr sogenannte Daten-CDs oder Daten-DVDs und diese Medien sind nicht bootbar.

Ladet euch das Image xxx-amd64-memstick.img und die entsprechende CHECKSUM-Datei herunter, und überprüft bitte zunächst das Image.<sup>35</sup> Stimmen die Checksum-Angaben überein, beschreibt ihr einen USB-Stick mit dem Image, falls ihr GNU/Linux verwendet, geht das wie folgt:

```
# dd if=<name-des-images> of=/dev/<name-des-USB-Stick-Devices> bs=4M
```

**Achtung: Geht dabei sorgsam vor, und schreibt nicht aus Versehen auf Eure Festplatte. Das Device erkennt ihr in der Regel, wenn ihr kurz nach dem Einstöpseln # dmesg eingibt, und euch die letzten Zeilen der Ausgabe anschaut.**

Stimmen die Input- und Output-Angaben überein, wurde der Stick ordentlich beschrieben und ist dann einsetzbar.

---

<sup>35</sup> Anm.: In der Regel werden sha512 Summen angegeben, und ihr braucht zum Überprüfen das – auch abhängig von eurem Betriebssystem - entsprechende Werkzeug, unter GNU/Linux zum Beispiel [sha512sum](#).

## 1.2 Der erste Start – FreeBSD und Hardware-Unterstützung

**Info Werkzeuge:** Es kommen [kbdmap](#)<sup>36</sup> und [dmesg](#)<sup>37</sup> zum Einsatz.

Stellt euren Rechner im BIOS nun bitte so ein, dass er von USB-Stick booten kann, alternativ bieten die meisten Systeme an, kurz nach dem Start ein Bootdevice-Menü über eine Taste aufrufen zu können, wählt dann den Stick aus. Ist der Stick in Ordnung, wird nach kurzer Zeit ein Bildschirm mit rotem FreeBSD-Logo und verschiedenen Bootoptionen angezeigt.

[freebsd\\_boot.png](#)

Mit [RETURN] startet ihr dann den eigentlichen Bootvorgang des Systems.

Es rasen eine Menge an Informationen über den Bildschirm und ihr landet dann an einer Auswahlbox mit Begrüßungstext. Hier könnt ihr wählen zwischen <INSTALL>, <SHELL> und <LIVE-CD>. Wählt bitte <LIVE-CD>, gebt an der Shell

```
# root
```

ein und drückt dann [Return], ihr könnt nun mit dem Live-Medium arbeiten.

Zunächst stellt ihr das Tastaturlayout auf Deutsch um, dazu ruft ihr das Programm [kbdmap](#) auf und wählt deutsches Layout (ob mit oder ohne Akzent-Tasten ist an dieser Stelle nicht so wichtig).

FreeBSD unterstützt verschiedene Architekturen,<sup>38</sup> ich beschränke mich hier allerdings nur auf die amd64-Architektur (64 Bit), bis auf wenige Ausnahmen sollte alles hier Geschriebene auch auf die i386-Architektur (32 Bit) zu übertragen sein.

Falls ihr mit GNU/Linux vertraut seit, kennt ihr vermutlich schon [dmesg](#)

... Text ...

---

36 Manpage <https://www.freebsd.org/cgi/man.cgi?query=kbdmap>

37 Manpage <https://www.freebsd.org/cgi/man.cgi?query=dmesg>

38 Siehe auch <https://www.freebsd.org/de/where.html>

## 1.3 Und nun auf eine Festplatte damit: Zur Installation

Na, Interesse bekommen, FreeBSD zu installieren? Falls ja, bootet wie schon oben erläutert mit eurem Stick, wählt aber diesmal bitte <INSTALL>. Hiermit startet ihr den Standard-Installer **bsdinstall**,<sup>39</sup> im folgenden gehe ich die einzelnen Schritte zur Installation des Basis-Systems durch.<sup>40</sup>

### 1.3.1 Das Tastatur-Layout

In der ersten Eingabemaske des Installers könnt ihr das Tastatur-Layout einstellen.

tastatur\_1.png

Standardmäßig ist US-Layout eingestellt. An dieser Stelle könnt ihr eine der drei deutschen Layout-Angebote wählen, eine Feinjustierung ist gegen Ende der Installation noch möglich. Das neue Layout könnt ihr hier auch testen, wichtig ist die korrekte Zuordnung von z, y, @, -, \_ u.s.f. Es kann sein, dass ä, ü, ö oder ß noch nicht funktionieren, das ist an dieser Stelle aber noch nicht so wichtig.<sup>41</sup>

tastatur\_2.png

### 1.3.2 Wie soll der Rechner heißen?

Im Folgeschritt geht es um die Vergabe eines Rechnernames, hier ist nicht nur der reine Hostname gemeint, sondern <hostname>.<domainname>.<sup>42</sup>

hostname\_1.png

Die Abbildung zeigt die Namesvergabe für meinen Desktop-PC **biber.unix.pc**. Habt ihr ein Netzwerk mit einem Domain-Namen, ist es sinnvoll, diesen hier auch einzusetzen.

### 1.3.3 Die Partitionierung

Das nächste Fenster des Installers bietet die Vorbereitung der Festplatte / Festplatten für die nachfolgende Installation an. Wer eine Suchmaschine zum Thema „richtige Partitionierung“ füttert, dürfte für einige Tage Lesestoff kredenzt bekommen. Es gibt sicherlich Festplattenaufteilungen, die als suboptimal zu bezeichnen sind, aber von groben Schnitzern abgesehen hängt es auch sehr von eigenen Vorlieben ab, wie partitioniert wird.

In den vergangenen Jahren bin ich unter GNU/Linux immer mehr dazu über gegangen, mit wenig Partitionen zu arbeiten.<sup>43</sup> Eigentlich reicht eine Partition für das Betriebssystem aus, Swap kann über ein Swapfile eingerichtet werden und ein regelmäßiges Backup der

---

39 Manpage <https://www.freebsd.org/cgi/man.cgi?bsdinstall>

40 Anm.: Falls ihr bisher mit GNU/Linux gearbeitet habt, könnt ihr in der Regel bei den dort verwendeten Installern ein komplettes Betriebssystem samt Xserver, Desktop-Oberflächen, Office, Browser, Mail und weiteren Programmen zur Installation auswählen. Dies ist bei **bsdinstall** anders, hiermit könnt ihr nur das Basis-System FreeBSD installieren.

41 Anm.: Zur Lokalisierung mit korrekter Tastatur-Konfiguration und **locale**-Einstellungen gibt es ein extra Unterkapitel.

42 Anm.: Für unser Heimnetzwerk habe ich die Domain **unix.pc** vergeben, die verschiedenen Rechner sind nach Tieren – bevorzugt Nagetieren – benannt.

43 Anm.: Damit meine ich wenige Partitionen pro System.

eigenen Dateien macht eine extra Partition für **/home** auch überflüssig, zumal in Fall unseres Heimnetzwerkes alle persönlichen Daten per NFS-Share auf die Desktop-Rechner eingebunden werden und von diesen Daten auf dem NAS einmal pro Nacht auch noch ein inkrementelles Backup erstellt wird.

Als ich im deutschsprachigen BSD-Forum meine erste Rohfassung dieser Dokumentation zur Diskussion gestellt habe, wurde mir auch zu möglichst wenig Partitionen auf einem Desktop-System geraten.<sup>44</sup>

Daher halte ich es auch unter FreeBSD so, mit möglichst wenig Partitionen (bzw. bei Verwendung des Dateisystems ZFS mit wenig Data-Sets) zu arbeiten.

Arbeitet ihr mit Windows, ist das Standarddateisystem wohl heutzutage nur noch ntfs, ob es überhaupt noch möglich ist, ein Windows 8 oder 10 unter vfat zu installieren, kann ich mangels Erfahrungen mit der Administration dieser Betriebssysteme nicht sagen.

Hattet ihr auch schon mit GNU/Linux zu tun, bietet sich eine ganz andere Situation: Was hätten' s denn gern für ein Dateisystem? ext3 / ext4, XFS, jfs, reiser, reiser4, btrfs? Seit Jahren nutze ich unter GNU/Linux das robuste ext4-Dateisystem, XFS und jfs habe ich aber auch schon ohne Schwierigkeiten verwendet. Meines Erachtens sind es heutzutage vor allem ext4, XFS und btrfs, die für eine Installation von GNU/Linux in Frage kommen, zumal an diesen Dateisystemen und Werkzeugen zur Wartung wird aktiv entwickelt.

Unter FreeBSD ist es noch etwas überschaubarer, eigentlich kommen nur zwei Dateisysteme in Frage, diese unterscheiden sich aber dermaßen voneinander, dass jedem dieser Dateisysteme ein eigenes Unterkapitel gewidmet ist.

### **1.3.3.1 Dateisysteme: Der Klassiker UFS2**

... Text ...

#### **1.3.3.1.1 UFS2 für Bequeme: Automatische Einrichtung durch den Installer**

Ihr habt kein großes Interesse daran, euch mit der Partitionierung zu befassen? Dann „kommt“ euch der Installer mit der Möglichkeit „entgegen“, dies automatisch erledigen zu lassen. Die Einstellungen für diese automatische Partitionierung sind in der Regel gut gesetzt.

... Text ...

#### **1.3.3.1.2 Manuelle Partitionierung einer Festplatte mit der Shell**

**Info Werkzeuge:** Es kommen **gpart**<sup>45</sup> und **newfs**<sup>46</sup> zum Einsatz.

Ihr habt spezielle Wünsche bei der Aufteilung der Festplatte oder ihr wollt diese Arbeit schon aus didaktischen Gründen gerne manuell mit einer Shell durchführen? Dann seid ihr bei diesem Unterkapitel genau richtig.

---

44 Siehe zum Beispiel <https://www.bsdforen.de/threads/freebsd-auf-dem-desktop-von-einem-einsteiger-f%C3%BCr-einsteiger-innen.32670/#post-281937>

45 Manpage <https://www.freebsd.org/cgi/man.cgi?gpart>

46 Manpage <https://www.freebsd.org/cgi/man.cgi?newfs>

Im folgenden beschreibe ich die manuelle Partitionierung einer SSD unter Verwendung des Dateisystems UFS2.

Es handelt sich bei dieser Beschreibung um die erste Platte, also um **/dev/ada0** (FreeBSD) bzw. **/dev/sda** (GNU/Linux), die zweite Platte wird unter FreeBSD mit **/dev/ada1**, unter Linux mit **/dev/sdb** bezeichnet. Vermutlich wollt auch ihr FreeBSD auf der ersten Platte installieren, falls nicht, passt bitte die Angaben entsprechend an.<sup>47</sup>

Beim Abschnitt zur Partitionierung wählt ihr die Shell aus und lasst euch zunächst mit

```
# gpart show -p ada0
```

Informationen zu eurer Festplatte anzeigen.

**Achtung: Beim folgenden Arbeiten mit gpart achtet genau auf die Eingaben, sonst droht ggf. ungewollter Datenverlust.**

Dann geht ihr, falls auf dieser Platte keine mehr für euch erhaltenswerten Daten sind, bitte wie folgt vor:

1. **# gpart destroy -F ada0**  
Damit löscht ihr die Partitionstabelle
2. **# gpart create -s gpt ada0**  
Damit legt ihr eine neue gpt Tabelle an
3. **# gpart add -t freebsd-boot -s 512k -a4k -l boot0 ada0**  
Hierdurch legt ihr eine 512 kb kleine 4k ausgerichtete auf **boot0** gelabelte Partition an
4. **# gpart bootcode -b /boot/pmbr -p /boot/gptboot -i 1 ada0**  
Schreiben des Bootcodes für FreeBSD
5. **# gpart add -t freebsd-swap -l swap0 -b 1m -s 8g ada0**  
Dies erzeugt eine 8 GB große Swap-Partition mit 1M Ausrichtung und dem Label swap0
6. **# gpart add -t freebsd-ufs -l system0 -a 1m -s 15g ada0**  
**# newfs -U -t /dev/gpt/system0**  
Hierdurch legt ihr eine 15 GB große UFS-Partition mit 1M Ausrichtung und dem Label system0 an, hierhin soll FreeBSD installiert werden. Die Partition formatiert ihr durch das Werkzeug **newfs** mit UFS2, mit **-t** schaltet ihr Trim ein,<sup>48</sup> mit **-U** Softupdates, ein Journaling der Softupdates (**-SUJ**) wird nicht aktiviert
7. **# gpart add -t freebsd-ufs -l daten0 -a 1m -s 45g ada0**  
**# newfs -U -t /dev/gpt/daten0**  
Damit legt ihr eine 45 GB große UFS-Partition mit 1M Ausrichtung und dem Label daten0 an, und formatiert sie mit UFS. Hierhin können persönliche Daten geschrieben werden. Punkt 7 ist nicht erforderlich für die Installation, falls euch eine UFS-Partition für das System und eure Daten reicht, passt bitte die Größe unter Punkt 6 eurer Festplattengröße entsprechend an und lasst den Punkt 7 aus.
8. **# mount /dev/gpt/system0 /mnt && mkdir /mnt/boot**  
Der Installer installiert nach **/mnt**, daher sind diese Schritte erforderlich.
9. **# exit**  
Nach dem Verlassen der Shell wird automatisch die Installation von FreeBSD gestartet

Falls an ZFS kein weiteres Interesse besteht, geht es mit **Kapitel 1.3.4** weiter.

---

<sup>47</sup> Anm.: Nehmt ihr die zweite Platte, ersetzt ihr bitte alle ada0-Angaben durch ada1.

<sup>48</sup> Anm.: Aktiviert **Trim** bitte nur bei bestimmten SSDs, die diese Technik unterstützen, bei einer mechanischen Festplatte lasst ihr das **-t** weg.

## 1.3.3.2 ZFS: Ein innovatives Dateisystem

... Text ...

### 1.3.3.2.1 ZFS für Bequeme: Automatische Einrichtung durch den Installer

Auch bei der Verwendung von ZFS braucht ihr nichts großartig manuell zu machen, der Installer bietet Euch eine automatische Einrichtung einer ZFS-Umgebung für FreeBSD an, bei der ihr allerdings einige Dinge modifizieren könnt.

### 1.3.3.2.2 Root-on-ZFS: Manuelle Einrichtung

#### 1.3.3.2.2.1 Erzeugen des Pools und der Data-Sets

**Info Werkzeuge:** Zum Einsatz kommen **gpart**, **zpool**<sup>49</sup> und **zfs**<sup>50</sup>

Der FreeBSD-Installer ist wie schon geschrieben in der Lage, eine ZFS-Umgebung automatisch zu kreieren für eine Root-on-ZFS Installation. Bei solchen Systemen gibt ein

```
# zfs list
```

eine große Anzahl verschiedener Data-Sets aus, die sicherlich gut vorkonfiguriert sind, allerdings ist mir diese Fülle von Data-Sets zu unübersichtlich und ähnlich wie im Kapitel zur manuellen Einrichtung einer Festplatte mit UFS2 reichen mir eigentlich neben der freebsd-boot-, der freebsd-swap- und der freebsd-zfs-Partition im ZFS-Pool ein Data-Set für das System und eines für Daten. Der FreeBSD-Installer erlaubt wie auch schon im Kapitel zu UFS2 gesehen beim Abschnitt zur Partitionierung eine komplett manuelle Einrichtung und ähnlich wie bei UFS2 beschreibe ich nun eine manuelle Root-on-ZFS Einrichtung am Beispiel meines Desktop-Rechners **biber.unix.pc**.

Die Einrichtung erfolgt auf einer 256 GB großen SSD, auf der neben einer freebsd-boot- und einer freebsd-swap- eine freebsd-zfs-Partition für einen Pool angelegt wird, ich verzichte also zunächst auf Redundanz, die Einrichtung erfolgt aber so, dass ich später durch den Einbau einer weiteren gleichgroßen SSD die Möglichkeit habe, den ZFS-Pool zu spiegeln.

Ich wähle dazu im Abschnitt zur Festplatteneinrichtung die <Shell> für die manuelle Partitionierung aus und führe folgende Schritte durch – falls die Platte neu ist, entfallen die Schritte 1 und 2:

**1. # zpool import -f -N <poolname>**

Hiermit wird ein eventuell vorhandener alter Pool importiert, die Option -N sorgt dafür, dass die Data-Sets dieses Pools nicht gemountet werden. Wenn keinerlei Daten mehr von diesem alten Pool benötigt werden, kann der Pool gelöscht werden.

**2. # zpool destroy -f <poolname>**

Dies zerstört den alten Pool vollständig.

**3. # gpart destroy -F ada0**

Dies zerstört das Geom ada0 samt Partitionstabelle.

**4. # gpart create -s gpt ada0**

Erzeugen einer GPT Partitionstabelle

---

49 Manpage <https://www.freebsd.org/cgi/man.cgi?zpool>

50 Manpage <https://www.freebsd.org/cgi/man.cgi?zfs>

5. **# gpart add -t freebsd-boot -l boot0 -a4k -s 512k ada0**  
Anlegen einer 512 KB großen Bootpartition mit korrekter Ausrichtung und der Labelvergabe **boot0**
6. **# gpart add -t freebsd-swap -l swap0 -s 8g ada0**  
Mein Rechner hat 8 GB Ram, die Größe der Swap-Partition von 16 GB ist Ansichtssache, sie erhält das Label **swap0**
7. **# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada0**  
Dies schreibt den Bootcode, wichtig ist hier die Angabe **/boot/gptzfsboot** statt **/boot/gptboot**, sonst bootet das System nicht
8. **# gpart add -t freebsd-zfs -l system0 -s 235g ada0**  
Die ZFS-Partition wird hier auf **system0** gelabelt, die Größe richtet sich nach dem verbleibenden Platz auf der SSD und sollte etwas kleiner als dieser verbleibende Platz gewählt werden
9. **# zpool create -m /zbiber zbiber /dev/gpt/system0**  
Hiermit wird ein Pool **zbiber** erzeugt für die mit **system0** gelabelte ZFS-Partition und auf das Verzeichnis **/zbiber** gemountet, die Meldung **cannot mount /zbiber: failed to create mountpoint...** kann ignoriert werden
10. **# zfs create -o mountpoint=none zbiber/ROOT**  
Dies erzeugt das Data-Set **zbiber/ROOT** ohne Mountpunkt
11. **# zfs create -o mountpoint=/mnt zbiber/ROOT/default**  
Dies erzeugt das Data-Set **zbiber/ROOT/default**, hierher erfolgt gleich die Installation des FreeBSD-Systems, da der Installer in den Pfad **/mnt** installiert, muss demnach der Mountpunkt temporär für die Installation auf **/mnt** gesetzt werden
12. **# zfs create -o mountpoint=/daten zbiber/daten**  
Dieses Data-Set möchte ich für diverse Daten wie den Inhalt von **/home** und auch für **poudriere** nutzen, es ist für die Installation entbehrlich, die Meldung **filesystem successfully created, but not mounted...** kann ignoriert werden
13. **# zfs set compression=lz4 zbiber/ROOT/default**  
Hier setze ich die Option **compression=lz4** für das Data-Set, wohin gleich FreeBSD installiert wird. Zum Setzen von Optionen und verschiedenen Kompressionsverfahren für Data-Sets siehe die Manpage zu **zfs**
14. **# zfs set copies=2 zbiber/daten**  
Hier wird für das Dataset **zbiber/daten** die Option **copies=2** gesetzt, jede Datei wird zweimal angelegt und somit Redundanz erzeugt. Achtung: Dies verbraucht zusätzlichen Speicherplatz.
15. **# zpool set bootfs=zbiber/ROOT/default zbiber**  
Hier wird das Verzeichnis festgelegt, von welchem gebootet werden soll, die Syntax ist:  
**zpool set bootfs=<Name-des-Datasets-von-Root> <Poolname>**  
Vor dem nächsten Schritt wird noch per **mount** geprüft, ob **zbiber/ROOT/default** auf **/mnt** gesetzt ist
16. **# mkdir -p /mnt/boot/zfs**  
Dies erzeugt das Bootverzeichnis, in das für den ZFS-Betrieb notwendige Sachen installiert werden
17. **# exit**  
Nach dem Verlassen der Shell wird automatisch die Installation von FreeBSD gestartet

Nach dem Durchlauf der Installation geht es zunächst mit den **Kapiteln 1.3.4 ff** weiter. Wenn ihr die Installerschritte bis zum Abschluss abgearbeitet habt, ist bei einer Root-on-ZFS-Installation noch ein wenig Nacharbeit nötig, die ich nun erläutern werde. Als Beispiel dient wieder mein Desktop-Rechner **biber.unix.pc**.

### 1.3.3.2.2 Letzte Schritte bei manueller Root-on-ZFS Einrichtung

**Info Werkzeuge:** Ihr braucht hierfür den Editor `ee`<sup>51</sup>

Vor dem Abschluss bietet euch der Installer noch an, per Shell in das neue System zu wechseln – dies nennt man auch `chroot`<sup>52</sup> – nach Bestätigung dieses Angebotes bei meiner Beispiel-Installation erscheint vor der Eingabeshell der Hostname `biber`, woran ich erkennen kann, dass ich nun ins installierte System gewechselt bin.

Ich ergänze mit dem Editor `ee` die Datei `/boot/loader.conf` um folgende Einträge:

```
# zfs_load="YES"
# vfs.root.mountfrom="zfs:zbiber/ROOT/default"
```

In die Datei `/etc/rc.conf` kommt:

```
# zfs_enable="YES"
# keymap=de.acc
```

Danach wird der PC komplett herunter gefahren und vor der weiteren Einrichtung noch ein letztes Mal mit dem Installationsmedium gebootet. Diesmal wird nicht der Installer, sondern die <Shell> gewählt, per

```
# kbdmap
```

deutsches Tastatur-Layout eingestellt und dann per

```
# zpool import -f -N zbiber
```

der neue Pool importiert, ohne die Data-Sets zu mounten. Das Data-Set mit der FreeBSD-Installation hat als Mountpunkt noch `/mnt`, das wird nun geändert, mit

```
# zfs set mountpoint=legacy zbiber/ROOT/default
```

wird der Mountpunkt auf `legacy` gesetzt.

Dann wird der Pool exportiert mit

```
# zpool export -f zbiber
```

danach kann per Reboot in das neue System gestartet werden.

Die folgenden Abschnitte des 1. Kapitels wurden bei einer manuellen Root-on-ZFS Einrichtungs schon abgearbeitet, weiter geht es daher mit **Kapitel 2**.

## 1.3.4 Die Paßwortvergabe für den Administrator

... Text ...

## 1.3.5 Auswahl einiger zu startender Dienste

... Text ...

---

51 Anm.: Ausführlicher gehe ich auf diesen unter FreeBSD standardmäßig installierten Editor später noch ein, die Bedienung ist simpel, mit `# ee /Pfad-zu-Datei` öffnet ihr die Datei, dann könnt ihr schreiben, mit [Esc] beendet ihr den Eingabemodus, mit [Return] wird der Editor verlassen (ist vorgewählt) und mit einem weiteren [Return] werden die Änderungen in die Datei geschrieben (ist vorgewählt), zu weiteren Befehlen siehe auch die Manpage <https://www.freebsd.org/cgi/man.cgi?query=ee>

52 Siehe <https://de.wikipedia.org/wiki/Chroot>

## 2 Die Konfiguration von FreeBSD

... Text ...

## 3 FreeBSD als Desktop-System

... Text ...

## 4 Fortgeschrittene Themen

... Text ...

### 4.1 FreeBSD im eigenen Netzwerk

... Text ...

### 4.2 Migration eines Dateiservers von GNU/Linux nach FreeBSD

... Text ...

### 4.3 Anlegen und pflegen eigener Repositorien

... Text ...

#### 4.3.1 Paketbau mit poudriere

**Info Werkzeuge:** Es kommen **poudriere**<sup>53</sup> und **jail**<sup>54</sup> zum Einsatz

Im folgenden Kapitel geht es mit **poudriere** um eines der mächtigsten Werkzeuge zum Paketbau aus den Ports. Ich setze eine FreeBSD-Installation auf ZFS voraus,<sup>55</sup> desweiteren halte ich es für sinnvoll, für poudriere ein eigenes Data-Set mit Redundanz anzulegen:

```
# zfs create -o mountpoint=/daten/poudriere zbiber/daten/poudriere
# zfs set copies=2 zbiber/daten/poudriere
```

Installiert euch nun bitte poudriere direkt über pkg oder aber baut es aus den Ports:

```
# cd /usr/ports/ports-mgmt/poudriere && make install clean
```

Nach der Installation richtet ihr **poudriere** ein, es gibt dazu einen lesenswerten Artikel im Wiki des deutschsprachigen BSD-Forums.<sup>56</sup>

Im folgenden beschreibe ich die Einrichtung auf meinem Desktop-Rechner **biber.unix.pc**, ich setze dafür voraus, dass es ein Data-Set für **poudriere** gibt, was auf den Pfad **/daten/poudriere** gemountet ist.

---

53 Manpage <https://www.freebsd.org/cgi/man.cgi?query=poudriere>

54 Manpage <https://www.freebsd.org/cgi/man.cgi?query=jail>

55 Anm.: Eine Verwendung auf UFS2 geht auch, bitte lest dazu den Wiki-Beitrag, falls ihr euch für eine Installation von FreeBSD auf UFS2 entschieden habt: <https://wiki.bsdforen.de/kategorie:howto:poudriere>

56 Siehe <https://wiki.bsdforen.de/kategorie:howto:poudriere>

Zunächst lege ich eine Konfigurationsdatei **/usr/local/etc/poudriere.conf** an:

```
ZPOOL=zbiber
ZROOTFS=/daten/poudriere
FREEBSD_HOST=http://www1.de.FreeBSD.org/
RESOLV_CONF=/etc/resolv.conf
BASEFS=/daten/poudriere/base
POUDRIERE_DATA=${BASEFS}/data
USE_PORTLINT=no
USE_TMPFS=yes
DISTFILES_CACHE=/daten/poudriere/distfiles
CHECK_CHANGED_OPTIONS=verbose
CHECK_CHANGED_DEPS=yes
# Path to the RSA key to sign the PKGNG repo with. See pkg-repo(8)
# PKG_REPO_SIGNING_KEY=/etc/ssl/keys/repo.key
PARALLEL_JOBS=4
PREPARE_PARALLEL_JOBS=4
WRKDIR_ARCHIVE_FORMAT=txz
NOLINUX=yes
ALLOW_MAKE_JOBS=yes
```

Bitte vergleicht diese Datei mit der **/usr/local/etc/poudriere.conf.sample**, sie ist ausführlich dokumentiert und dort findet ihr gute Erklärungen der einzelnen Einträge. Eventuell passt ihr eure eigene poudriere.conf abweichend von meinem Beispiel anhand dieser Beispieldatei auch noch nach euren Wünschen an.

Bitte sorgt nun zunächst dafür dass Euer Ports-Tree aktuell ist, dazu führt ihr am besten folgenden Befehl aus:

```
# portsnap auto
```

Und nun macht ihr poudriere mit den aktuellen Ports „bekannt“:<sup>57</sup>

```
# poudriere ports -c
```

---

57 Anm.: Hiermit liest **poudriere** die Ports lediglich aus, es findet dadurch keine Aktualisierung des Ports-Baumes statt.