

## Programmieraufgaben zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme

### Vorbereitung zu den Programmieraufgaben

(20. April – 24. April 2015)

## 1 Übersicht

### VServer und Versionsverwaltung (SVN)

Im Rahmen der Programmieraufgaben stellen wir jedem zur Vorlesung angemeldeten Studierenden jeweils einen virtuellen „Rootserver“ (VM) mit eigener IPv4-Adresse und eigenem IPv6-Subnetz zur Verfügung. Die VMs dienen als einheitliche Testplattform, auf denen Ihre Abgaben kompilierbar und ausführbar sein müssen. Der Zugang zu den VMs erfolgt ausschließlich über SSH (s. Abschnitt 2.3).

Da die Bereitstellung von mehr als 700 VMs eine beträchtliche Anzahl echter Hardware zur Virtualisierung erfordert, haben wir uns zu folgenden beiden Schritten entschieden:

- VMs müssen einmalig angefordert werden (s. Abschnitt 2.2).
- Inaktive VServer werden nach 25 h abgeschaltet (PowerOff<sup>1</sup>) und müssen neu gestartet werden.

Die Anforderung bzw. das Wiedereinschalten von VServern geschieht vollautomatisch und innerhalb weniger Minuten.

Die Abgabe Ihrer Programmieraufgaben (ebenso wie das Anfordern der VServer) geschieht über Subversion (SVN, s. Abschnitt 2.1). Dabei handelt es sich um eines der gängigen Versionsverwaltungssysteme, welche in der Softwareentwicklung eingesetzt werden. Es ermöglicht mehreren Programmierern gemeinsam an einem Softwareprojekt zu arbeiten. Quelltexte werden in einem **Repository** abgelegt, welches auf einem Server bereitliegt. Von dort kann der Inhalt des Repositories ausgecheckt werden, so dass eine lokale Kopie vorliegt. Auf der Kopie kann nun gearbeitet werden. In regelmäßigen Abständen sollte ein **commit** durchgeführt werden, welcher Änderungen auf den Server repliziert. Dabei werden im Repository lediglich die Änderungen gegenüber der letzten Revision gespeichert. Andere Benutzer können diese Änderungen mittels eines **update** herunterladen und so ihre lokale Kopie aktualisieren. Ausführliche Informationen über Subversionen finden Sie in [1, 2].

### Warum brauchen wir SVN?

Es gibt mehrere Gründe, weswegen wir für die Programmieraufgaben SVN verwenden:

1. Jeder Informatiker sollte mit wenigstens einem Versionsverwaltungssystem umgehen können. Falls Sie es nicht schon längst können, sollten Sie es besser früher als später lernen.
2. Wir brauchen ein System, wie Sie Ihre Programme abgeben können. Anstatt uns Ihre Programme via Email zu schicken oder als Tarball über ein Webformular hochzuladen, müssen Sie nur sicherstellen, dass Ihre aktuelle Version im SVN liegt.
3. Es ermöglicht uns, Ihnen auf elegante Art die Vorlesungsunterlagen bereitzustellen: Sie müssen lediglich die lokale Kopie Ihres SVN aktualisieren (ein Befehl oder zwei Klicks) und nicht jeden Downloadlink einzeln anklicken. Außerdem sehen Sie auf einen Blick, was sich wann geändert hat.

### Ich will aber Git verwenden!

Mach doch. Nennt sich git-svn.

<sup>1</sup> Es gibt keine Snapshots, d. h., nicht gespeicherte Daten gehen verloren.

## Wo bekomme ich SVN?

- Linux-Nutzer installieren svn mit der Paketverwaltung ihrer Wahl.
- Nutzer von OS X erhalten SVN entweder über XCode CLI-Tools [3] oder über MacPorts bzw. Fink.
- Windows-Nutzer wollen sicherlich einen grafischen Client. Empfehlenswert ist hier TortoiseSVN[4], welcher sich direkt in das Kontextmenü des Windows-Explorers integriert.

## 2 Wie komme ich auf meine VM?

### 2.1 SVN Repository Auschecken

Zunächst müssen Sie Ihr Arbeitsverzeichnis für die Programmieraufgaben auschecken:

#### Kommandozeile

```
~ $ svn co --username <LRZ-Kennung> https://vcs.net.in.tum.de/svn/grnvss15 <Zielverzeichnis>
```

#### TortoiseSVN

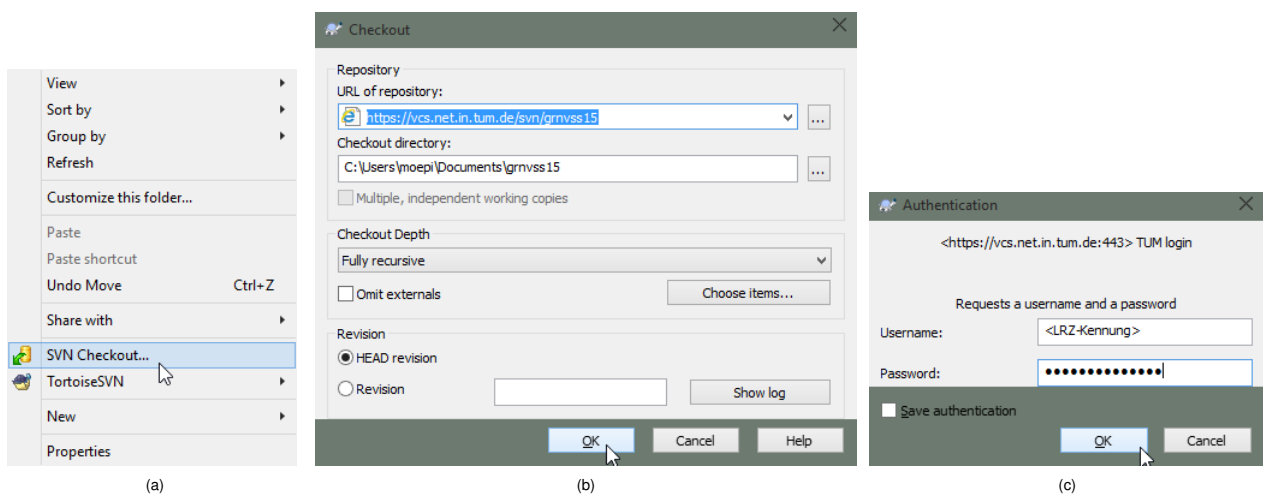


Abbildung 1: (a) Rechtsklick im Explorer, dann „SVN Checkout“ auswählen. (b) Geben Sie die URL zum SVN an. (c) Tragen Sie Ihre 7-stellige LRZ-Kennung und das zugehörige Passwort ein.

Folgen Sie der Anleitung in Abbildung 1. Anschließend sollten Sie die folgende Verzeichnisstruktur haben:

```
| - bookscans
|   | - init
| - pub
|   | - altklausuren
|   |   | - (...)
|   | - slides_chap0.pdf
|   | - slides_chap1.pdf
|   | - tutorgruppen-tutoren_mapping.txt
| - users
|   | - <LRZ-Kennung>
|   |   | - grades.txt
|   |   | - vm-bitte.txt
```

Neben den bereits bekannten Verzeichnissen pub und bookscans haben Sie nun Ihr Arbeitsverzeichnis users/<LRZ-ID> erhalten. Darin befinden sich zwei Textdateien

- vm-bitte.txt (mehr dazu gleich) und
- grades.txt, in der Sie die Bewertungen zu Ihren Programmieraufgaben erhalten werden.

**Wichtig:** Sie haben in Ihrem Arbeitsverzeichnis keine Berechtigungen, neue Dateien oder Verzeichnisse anzulegen. Sie können lediglich die Datei vm-bitte.txt modifizieren (oder sogar löschen und in diesem Fall auch neu erstellen). Für die einzelnen Programmieraufgaben werden wir Ihnen zu gegebener Zeit Unterverzeichnisse assignmentX erzeugen, innerhalb derer Sie auch Dateien und Verzeichnisse anlegen können werden.

## 2.2 VM anfordern

Wie eingangs erwähnt müssen Sie Ihre VM zunächst anfordern (bzw. einschalten). Dies geschieht dadurch, dass Sie die Datei `vm-bitte.txt` in Ihrem Arbeitsverzeichnis

1. verändern (es muss lediglich eine Änderung sein und die Textdatei muss danach größer 0 B sein) und
2. die modifizierte Datei danach committen.

Sobald Ihre VM gestartet ist (eine Frage weniger Minuten) wird in Ihrem Arbeitsverzeichnis die Datei `vm-status.txt` aktualisiert. Um diese zu erhalten, müssen Sie ihr SVN updaten.

### Kommandozeile

```
~/grnvss15/users/<LRZ-Kennung> $ date >> vm-status && svn commit -m "VM angefordert"  
~/grnvss15/users/<LRZ-Kennung> $ sleep 60  
~/grnvss15/users/<LRZ-Kennung> $ svn update
```

### TortoiseSVN

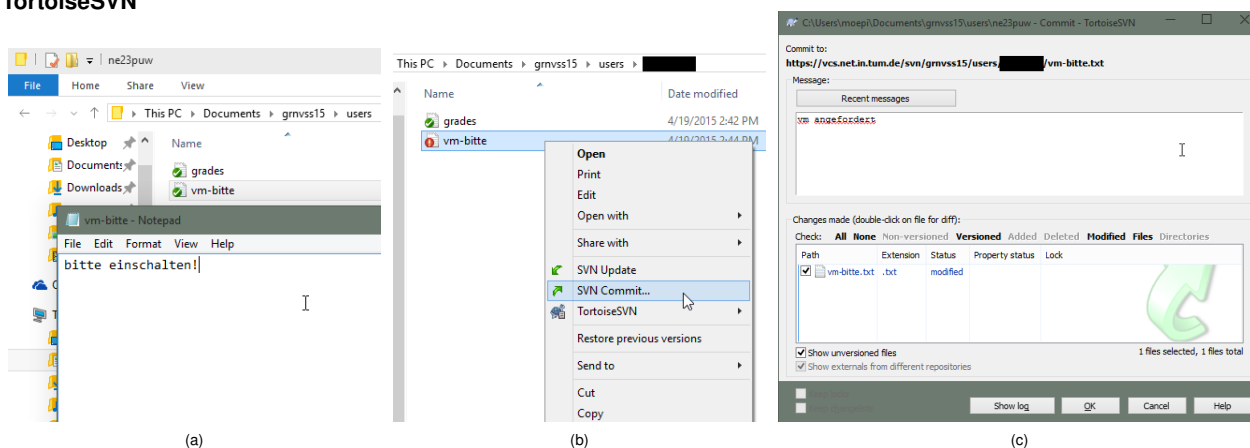


Abbildung 2: (a) Editieren und speichern Sie die Datei `vm-bitte.txt`.

(b) Das rote Ausrufezeichen im Dateisymbol weist darauf hin, dass die Datei lokale Änderungen enthält, die noch nicht auf den Server committen wurden. Mittels Rechtsklick und „SVN-Commit“ senden Sie die Änderungen zum Server.

(c) Jeder Commit sollte eine kurze Beschreibung der Änderungen enthalten. Klicken Sie anschließend auf „OK“ um die Änderungen hochzuladen. Das Dateisymbol von `vm-bitte.txt` sollte nun wieder grün sein (evtl. müssen Sie die Ansicht aktualisieren).

Der Inhalt Ihres Arbeitsverzeichnisses sollte sich nun verändert haben:

```
| - (...)  
| - users  
|   | - <LRZ-Kennung>  
|     | - fingerprints.txt  
|     | - grades.txt  
|     | - hostname.txt  
|     | - id_rsa  
|     | - id_rsa.pub  
|     | - known_hosts  
|     | - password.txt  
|     | - vm-bitte.txt  
|     | - vm-status.txt
```

- `fingerprints.txt` enthält die Fingerprints Ihres RSA-Schlüssels sowie des Hostkeys Ihrer VM.
- `hostname.txt` enthält den Hostnamen<sup>2</sup>, unter dem Sie Ihre VM erreichen.
- `id_rsa` und `id_rsa.pub` ist ein RSA-Schlüsselpaar, mit dem Sie Zugang zur Ihrer VM erhalten. Halten Sie den privaten Schlüssel `id_rsa` geheim.

**Wichtig:** Wenn Sie mit Linux oder OS X arbeiten, müssen Sie die Berechtigungen auf dem privaten Schlüssel korrigieren:

```
~/grnvss15/users/<LRZ-Kennung> $ chmod 600 id_rsa
```

Unter Windows ist das egal.

<sup>2</sup>Eigentlich ist es ein Fully Qualified Domain Name, wie wir noch lernen werden

- `vm-status.txt` enthält die SVN-Revision und die zugehörige Aktion auf Ihrer VM (start/stop).
- `password.txt` enthält das Root-Kennwort Ihrer VM, das Sie aber weder benötigen noch verwenden sollten.

## 2.3 Login auf der VM

Der Benutzername für die VMs ist `root`. Passwort benötigen Sie keines, sofern Sie das RSA-Schlüsselpaar verwenden. Andernfalls finden Sie das Passwort in der Datei `password.txt` in Ihrem Arbeitsverzeichnis. Die Adresse, unter der Sie Ihre VM erreichen, finden Sie in der Datei `hostname.txt`.

### Kommandozeile

```
~ $ ssh -i ~/grnvss15/users/<LRZ-Kennung>/id_rsa root@svm<xzy>.net.in.tum.de
```

Sollten Sie die Meldung „WARNING: UNPROTECTED PRIVATE KEY FILE!“ erhalten, haben Sie die Berechtigungen Ihres privaten Schlüssels nicht korrigiert (s. oben).

### Windows + MobaXterm

Da Microsoft es bis einschließlich Windows 10 nicht geschafft hat, seine Betriebssysteme wenigstens mit einem SSH-Client auszustatten<sup>3</sup>, müssen wir hier leider einige Umwege gehen. Der wahrscheinlich mit Abstand angenehmste und übersichtlichste Weg ist, *MobaXterm* [5] zu verwenden. Dabei handelt es sich um einen SSH-Client mit Unterstützung für SSHFS, integriertem Dateibrowser und einem XServer (den wir nicht brauchen). Es ist kostenfrei erhältlich und es gibt angenehmer Weise sogar eine Standalone-Binary (keine Installation notwendig). Nach der Installation folgen Sie bitte den Hinweisen in Abbildung 3a.

## 2.4 Arbeiten auf der VM

Es empfiehlt sich, direkt auf der VM zu programmieren. Dazu ist es notwendig, das SVN auf der VM direkt auszuchecken. Der Prozess ist in Abschnitt 2.1 (Kommandozeile) beschrieben.

Da vermutlich nicht jeder mit den kommandozeilen-basierten Texteditoren wie `vim` oder `nano` umgehen kann oder will, sollten Sie im Anschluss Ihr Homeverzeichnis auf der VM lokal mounten:

- Linux-Nutzer installieren sich SSHFS über ihren bevorzugten Paketmanager.
- OS X-Nutzer haben es leider etwas schieriger. Aber auch hier gibt es SSHFS-Clients. Eine sinnvolle Anleitung finden Sie unter [6]
- Windows-Nutzer brauchen garnichts mehr zu tun, da MobaXterm bereits einen SSHFS-Client mitbringt.

### Kommandozeile

Unter Linux

```
~ $ sshfs -i ~/grnvss15/users/<LRZ-Kennung>/id_rsa root@svm<xzy>.net.in.tum.de: -o allow_other,reconnect
```

und unter OS X

```
~ $ sshfs root@svm<xzy>.net.in.tum.de: -o IdentityFile=~/.grnvss15/users/<LRZ-Kennung>/id_rsa
```

Alternative können Sie natürlich auch auf einer lokalen Kopie des SVN-Repositories arbeiten. Allerdings müssen Sie es dann jedes mal, wenn Sie es auf der VM testen wollen, zuerst committen und die zweite Kopie des Repositories auf der VM updaten. Selbstverständlich können Sie auch Ihren eigenen Rechner zur Entwicklung verwenden. Allerdings müssen Sie sicher stellen, dass die Abgabe am Ende auf den VMs lauffähig ist. Außerdem benötigen Sie Linux. Unter OS X und FreeBSD unterscheiden sich zumindest Name und Pfad einiger Header. Unter Windows wären die Änderungen deutlich umfangreicher, weswegen wir dringend davon abraten.

## 3 Wie gehts jetzt weiter?

Machen Sie sich bitte mit SVN und Ihrer VM vertraut. Versuchen Sie, Dateien auf die VM zu übertragen und ein einfaches „Hello World“ in einer Sprache Ihrer Wahl zu schreiben. Bei Problemen Besuchen Sie bitte eine der Programmierübungen:

Bitte beachten Sie zu den Programmierübungen folgende Regeln:

1. Die Programmierübungen beginnen am Dienstag, den 21. April, und finden fortan **nur dann statt, wenn gerade eine Programmieraufgabe zu bearbeiten ist**.
2. Bitte kommen Sie nur mit konkreten Fragen oder Problemen zu den Programmier Tutoren – sie sind kein First-Level-Support bei Computerproblemen.

Die Veröffentlichung der ersten Programmieraufgabe ist für Montag, den 27. April, geplant.

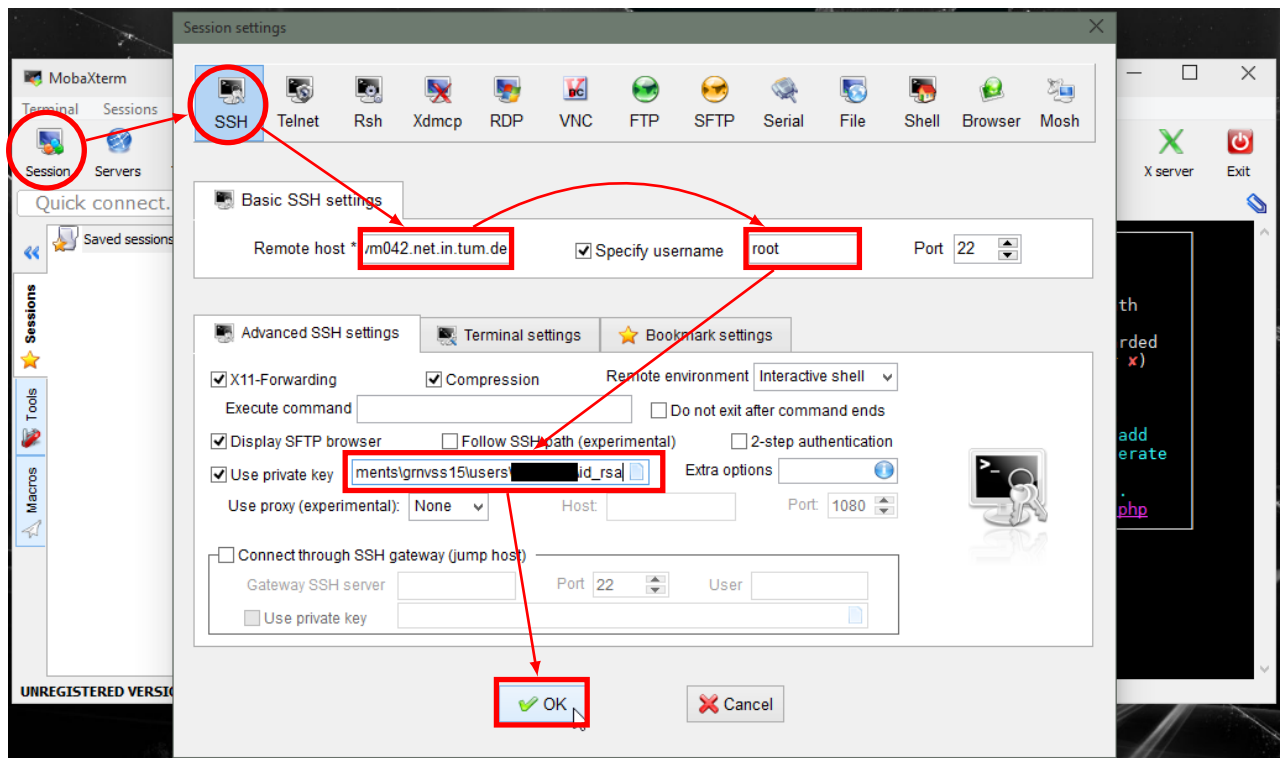
<sup>3</sup>Den Telnet-Client haben sie auch entfernt. Der ist aber wenigstens optional wieder installierbar.

Gruppe	Wochentag	Uhrzeit	Raum	Tutor
Mo-P1	Montag	12:00 – 14:00	03.07.023	Richard von Seck
Di-P1	Dienstag	12:00 – 14:00	00.08.059	Richard von Seck
Mi-P1	Mittwoch	12:00 – 14:00	03.07.023	Markus Ongyerth
Do-P1	Donnerstag	12:00 – 14:00	00.08.059	Markus Ongyerth

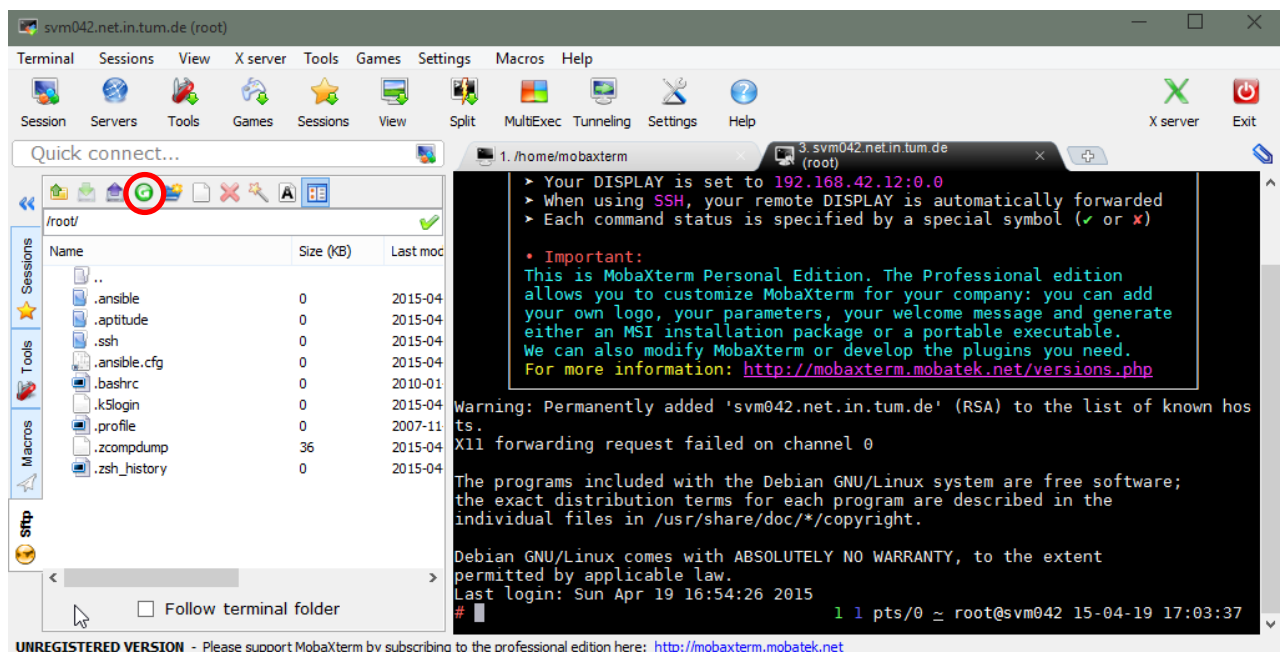
Tabelle 1: Programmiergruppen

## Literatur

- [1] *TortoiseSVN – FAQ*. <http://subversion.apache.org/faq.html>.
- [2] *TortoiseSVN – Manual*. <http://svnbook.red-bean.com/>.
- [3] *XCode CLI-Tools – Download*. [https://developer.apple.com/library/ios/technotes/tn2339/\\_index.html](https://developer.apple.com/library/ios/technotes/tn2339/_index.html).
- [4] *TortoiseSVN – Download*. <http://tortoisesvn.net>.
- [5] *MobarXterm – Download*. <http://mobaxterm.mobatek.net>.
- [6] T. Kessler, *How to mount a remote system as a drive using SSH in OS X*. <http://www.macissues.com/2014/10/13/how-to-mount-a-remote-system-as-a-drive-using-ssh-in-os-x/>.



(a)



(b)

Abbildung 3: (a) Tragen Sie als Remote Host den Namen Ihrer VM ein. Der Benutzername lautet root. Geben Sie als „Private Key“ den privaten RSA-Schlüssel an (Pfad zur Datei id\_rsa). Nachdem Sie auf „Ok“ geklickt haben, sehen Sie in der linken Hälfte des Fensters die eben angelegte Session. Ein Doppelklick darauf stellt die Verbindung zur VM her.

(b) Im rechten Teil des Fensters sehen Sie nun die Konsolensitzung. Im linken Teil des Fensters sehen Sie einen Dateibrowser, der Ihnen Zugriff auf das Dateisystem Ihrer VM ermöglicht.

**Hinweis:** Wenn Sie Dateien auf der Konsole angelegt oder gelöscht haben, müssen Sie unter Umständen Die Dateiansicht des Browsers aktualisieren. Danke roter Kreis.