

Tomcat auf ubuntu (athanassiou.me)

1 Inhaltsverzeichnis

<u>1</u>	<u>INHALTSVERZEICHNIS</u>	<u>1</u>
<u>2</u>	<u>TOMCAT INSTALLIEREN.....</u>	<u>2</u>
2.1	USER TOMCAT ANLEGEN	2
2.2	LINUX-BUILD FÜR TOMCAT 10 DOWNLOADEN	2
2.3	INSTALLATION AUSPACKEN	2
<u>3</u>	<u>ERSTELLEN EINES SYSTEMD DIENSTES</u>	<u>3</u>
3.1	SYSTEMD UND SYTEMCTL	3
3.2	JAVA VERZEICHNIS MERKEN	3
3.3	TOMCAT.SERVICE DATEI ERSTELLEN	3
3.4	TOMCAT.SERVICE - DER CODE.....	4
<u>4</u>	<u>TOMCAT MIT SYTEMCTL</u>	<u>5</u>
4.1	STARTEN, RESTARTEN UND STOPPEN	5
4.2	STATUS	5
<u>5</u>	<u>TOMCAT-USER ANLEGEN</u>	<u>5</u>
<u>6</u>	<u>... LETZTE SCHRITTE</u>	<u>6</u>

2 Tomcat installieren

2.1 User tomcat anlegen

Aus Sicherheitsgründen sollte Tomcat unter einem separaten, nicht privilegierten Benutzer ausgeführt werden. Das folgende Kommando erzeugt einen solchen User namens Tomcat:

```
$ sudo useradd -m -d /opt/tomcat -U -s /bin/false tomcat
```

Mit `/bin/false` als Standards-Shell des Benutzers stellt man sicher, dass es nicht möglich ist, sich als Tomcat anzumelden.

2.2 Linux-Build für Tomcat 10 downloaden

In einem `/tmp` Verzeichnis lädt man das neueste Tomcat-build runter und packt es aus. Der Befehl `wget` lädt Ressourcen aus dem Internet herunter.

```
$ cd /tmp
```

```
wget https://dlcdn.apache.org/tomcat/tomcat-10/  
v10.1/bin/apache-tomcat-10.1.tar.gz
```

2.3 Installation auspacken

Der Download wird mit dem Kommando `tar` entpackt:

```
$ sudo tar xzvf apache-tomcat-10*.tar.gz -C /opt/tomcat --  
strip-components=1
```

Da es einen User `Tomcat` gibt, kann er jetzt auch der `Owner` der extrahierten Installation werden:

```
$ sudo chown -R tomcat:tomcat /opt/tomcat/
```

```
$ sudo chmod -R u+x /opt/tomcat/bin
```

3 Erstellen eines systemd Dienstes

systemd ist **der** System- und Sitzungs-Manager (Init-System) von ubuntu/linux/unix, der für die Verwaltung **aller** auf dem System laufenden Dienste über die gesamte Betriebszeit des Rechners, vom Startvorgang bis zum Herunterfahren, zuständig ist.

3.1 systemd und sytemctl

Das Werkzeug, um systemd auf der Kommandozeile bzw. im Terminal zu verwalten, hört auf den Namen systemctl. Mit Hilfe von systemctl können Befehle an systemd gesendet werden, z.B. zum Steuern von Units, zum Abfragen des Status, zum Herunterfahren des Systems, zum Hochfahren, zum automatischen Hochfahren im Fehlerfall, etc..

systemd holt alle seine Vorgaben und Einstellungen zur Verwaltung aus Dateien, in der Terminologie von systemd sind dies "Units". Mit `$ systemctl list-units -all` kann man die "Units" von systemd einsehen. Die „Unit“-Dateien werden in dem Verzeichnis `etc/systemd/system` angelegt.

Der nächste Installationsschritt ist also das „Bekanntmachen“ von Tomcat beim systemd mittels systemctl, sprich das Erstellen einer entsprechenden „Unit“ Datei.

3.2 Java Verzeichnis merken

Was man griffbereit haben sollte, ist das Verzeichnis wo sich die Java Runtime sich befindet. Das Verzeichnis kann man nachschlagen mit:

```
$ sudo update-java-alternatives -l
```

Das Ergebnis wird so ähnlich aussehen wie:

```
java-1.11.0-openjdk-amd64 1111 /usr/lib/jvm/java-1.11.0-openjdk-amd64
```

3.3 tomcat.service Datei erstellen

Die Eigenschaften der Tomcat „Unit“ werden in der Datei `tomcat.service` definiert. Sie muss sich im Verzeichnis `/etc/systemd/system` befinden.

Die Datei kann man erstellen mit:

```
$ sudo nano /etc/systemd/system/tomcat.service
```

Nano ist ein einfacher, auf ubuntu vorinstallierter Editor. Er ermöglicht die Bearbeitung von Dateien in einem Terminal bzw. auf der Konsole, auch ohne grafische Umgebung. Nano hat einen geringeren Funktionsumfang als z.B. VIM oder Emacs, ist dafür aber einfacher zu bedienen. Nano eignet sich für kleinere, schnelle Bearbeitungen oder für das Schreiben von kurzen Skripten.

3.4 tomcat.service - der Code

In die obige (noch leere) Datei `/etc/systemd/system/tomcat.service` kommen jetzt folgende Zeilen hinein:

```
[Unit]
Description=Tomcat
After=network.target

[Service]
Type=forking

User=tomcat
Group=tomcat

Environment="JAVA_HOME= /usr/lib/jvm/java-1.11.0-openjdk-amd64"
Environment="JAVA_OPTS=-Djava.security.egd=file:///dev/urandom"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
Environment="CATALINA_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC"

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

RestartSec=10
Restart=always

[Install]
WantedBy=multi-user.target
```

Was heißt das jetzt im Einzelnen?

Environment:

- Einige Umgebungsvariablen werden festgelegt, um das Basisverzeichnis zu definieren
- Der **hervorgehobene** Wert von `JAVA_HOME` sollte den zuvor notierten Pfad (2.1) zum JRE beinhalten.

ExecStart / ExecStop:

- Mit diesem `serviced` kann man Tomcat starten / stoppen / restarten, indem der `serviced` die von der Installation bereit gestellten Start- und Herunterfahrskripts aufruft.

CATALINA_OPTS:

- Der Arbeitsspeicher, den die Java-VM zuweisen kann, wird begrenzt.

Restart:

- Bei einem Fehler wird der Tomcat-Dienst automatisch neu gestartet.

Der `systemd` Daemon muss nun noch durchgestartet werden, um die die neue „Unit“ zu laden:

```
$ sudo systemctl daemon-reload
```

4 Tomcat mit sytemctl

4.1 Starten, restarten und stoppen

Jetzt kann Tomcat mit:

```
$ systemctl stop tomcat  
$ systemctl start tomcat  
$ systemctl restart tomcat
```

gestartet, gestoppt und durchgestartet.

4.2 Status

Der Status kann mit:

```
$ systemctl status tomcat
```

erfragt werden. Die Ausgabe sieht ungefähr so aus:

- tomcat.service - Tomcat
Loaded: loaded (/etc/systemd/system/tomcat.service;)
Active: active (running) since Fri 2022-03-11 14:37:10 UTC; 2s ago
Process: 4845
ExecStart=/opt/tomcat/bin/startup.sh(code=exited status=0/ SUCCESS)
Main PID: 4860 (java)
Tasks: 15 (limit: 1132)
Memory: 90.1M
CGroup: /system.slice/tomcat.service
└─4860 /usr/lib/jvm/java-1.11.0-jdk.amd64/bin/java/logging.file

Mit `q` kann man die Ausgabe beenden.

5 Tomcat-User anlegen

Falls man die Manager-/Admim GUIs für nutzen möchte, sollte man die Nutzer und Passworte in `/opt/tomcat/conf/tomcat-users.xml` anlegen – das hängt aber von den persönlichen Präferenzen ab.

Wenn man den Zugriff auf den ganzen Server von außen nur mit einem Passwort zulassen will, macht man das ebenfalls in `/opt/tomcat/conf/tomcat-users.xml`:

```
$ nano /opt/tomcat/conf/tomcat-users.xml
```

öffnet die Datei, in der muss dann ein Eintrag wie folgender erfolgen:

```
<!-- Passwortschutz für Webapplikationen -->  
  
<role rolename="protected"/>  
<user username="name" password="passwd" roles="protected"/>
```

Falls der Zugriff auf einzelne webapps geregelt werden soll, so muss das in den `web.xml` Dateien der jeweiligen Anwendung erfolgen.

6 ... letzte Schritte

Jetzt kann man noch dafür sorgen das Tomcat bei einem Neustart des Servers auch neu gestartet wird:

```
$ sudo systemctl enable tomcat
```

und für den Fall der Fälle der Port 8080 freigeben:

```
$ sudo ufw allow 8080
```