Comment installation Tomcat sur debian 10

Introduction

Apache Tomcat est un serveur Web et un conteneur de servlets utilisés pour desservir les applications Java. Tomcat est une implémentation open source des technologies Java Servlet et JavaServer Pages, publiée par Apache Software Foundation. Ce tutoriel couvre l’installation de base et une partie de la configuration de la dernière version de Tomcat 9 sur votre serveur Debian 10.

Conditions préalables

Avant de commencer avec ce guide, vous devez avoir un utilisateur non root avec les privilèges + sudo + configurés sur votre serveur. Vous pouvez apprendre à le faire en complétant notre [Debian 10 Guide de configuration du serveur](https://www.digitalocean.com/community/tutorials/initial-server-setup-with-debian-10).

Étape 1 - Installez Java

Tomcat exige que Java soit installé sur le serveur pour que tout code d’application Web Java puisse être exécuté. Nous pouvons satisfaire cette exigence en installant OpenJDK avec apt.

Commencez par mettre à jour votre index de paquet apt:

sudo apt update

Puis installez le paquet Java Development Kit avec apt:

sudo apt install default-jdk

Maintenant que Java est installé, nous pouvons créer un utilisateur + tomcat +, qui sera utilisé pour exécuter le service Tomcat.

Étape 2 - Créer un utilisateur Tomcat

Pour des raisons de sécurité, Tomcat doit être exécuté en tant qu’utilisateur non privilégié (c.-à-d. pas \* racine \*). Nous allons créer un nouvel utilisateur et un nouveau groupe qui exécutera le service Tomcat.

Tout d’abord, créez un nouveau groupe \* tomcat \*:

sudo groupadd tomcat

Ensuite, créez un nouvel utilisateur \* tomcat \*. Nous ferons de cet utilisateur un membre du groupe \* tomcat \*, avec un répertoire personnel ` / / opt / tomcat + `(où nous installerons Tomcat), et un shell avec + / bin / false + `(donc personne ne peut se connecter au compte):

sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat

Maintenant que notre utilisateur \* tomcat \* est configuré, téléchargez et installons Tomcat.

Étape 3 - Installez Tomcat

Le meilleur moyen d’installer Tomcat 9 est de télécharger la dernière version binaire, puis de la configurer manuellement.

Retrouvez la dernière version de Tomcat 9 à la page [Tomcat 9 [Téléchargements](https://tomcat.apache.org/download-90.cgi)]. Au moment de la rédaction, la dernière version est \* 9.0.27 \*, mais vous devez utiliser une version stable ultérieure si celle-ci est disponible. Sous la section \* Distributions binaires \*, puis sous la liste \* Core \*, copiez le lien dans le fichier «tar.gz».

Revenez ensuite à votre session SSH et accédez au répertoire + / tmp + sur votre serveur. C’est un bon répertoire pour télécharger des éléments éphémères, comme l’archive Tomcat, dont nous n’aurons plus besoin après avoir extrait le contenu de Tomcat:

cd /tmp

Nous allons utiliser l’outil de ligne de commande + curl + pour télécharger l’archive. Installez + curl +:

sudo apt install curl

Maintenant, utilisez + curl + pour télécharger le lien que vous avez copié depuis le site Web de Tomcat:

curl -O

Nous allons installer Tomcat dans le répertoire + / opt / tomcat +. Créez le répertoire, puis extrayez l’archive avec les commandes suivantes:

sudo mkdir /opt/tomcat

sudo tar xzvf apache-tomcat-9\*tar.gz -C /opt/tomcat --strip-components=1

Ensuite, nous allons configurer les autorisations utilisateur appropriées pour notre installation.

Étape 4 - Mettre à jour les autorisations

L’utilisateur \* tomcat \* que nous avons créé doit avoir accès à l’installation de Tomcat. Nous allons le configurer maintenant.

Accédez au répertoire dans lequel nous avons décompressé l’installation de Tomcat:

cd /opt/tomcat

Attribuez la propriété du groupe \* tomcat \* à l’ensemble du répertoire d’installation:

sudo chgrp -R tomcat /opt/tomcat

Ensuite, donnez au groupe \* tomcat \* + read + accès au répertoire + conf + et à tout son contenu, et + execute + accès au répertoire lui-même:

sudo chmod -R g+r conf

sudo chmod g+x conf

Faites de l’utilisateur \* tomcat \* le propriétaire des répertoires + webapps,` + work`, + temp et` + log`:

sudo chown -R tomcat webapps/ work/ temp/ logs/

Maintenant que les autorisations appropriées sont configurées, nous allons créer un fichier de service systemd pour gérer le processus Tomcat.

Étape 5 - Créer un fichier de service systemd

Nous voulons pouvoir exécuter Tomcat en tant que service. Nous allons donc configurer le fichier de service systemd.

Tomcat a besoin de savoir où Java est installé. Ce chemin est communément appelé + JAVA\_HOME +. Le moyen le plus simple de rechercher cet emplacement consiste à exécuter cette commande:

sudo update-java-alternatives -l

Outputjava-1.11.0-openjdk-amd64 1111

Votre + JAVA\_HOME + est la sortie de la dernière colonne (mise en évidence ci-dessus). Dans l’exemple ci-dessus, le correct + JAVA\_HOME + pour ce serveur serait:

JAVA\_HOME/usr/lib/jvm/java-1.11.0-openjdk-amd64

Votre + JAVA\_HOME + peut être différent.

Avec cette information, nous pouvons créer le fichier de service systemd. Ouvrez un fichier appelé + tomcat.service dans le répertoire` + / etc / systemd / system` en tapant:

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

Collez le contenu suivant dans votre fichier de service. Modifiez la valeur de + JAVA\_HOME + si nécessaire pour qu’elle corresponde à la valeur trouvée sur votre système. Vous pouvez également vouloir modifier les paramètres d’allocation de mémoire spécifiés dans + CATALINA\_OPTS +:

/etc/systemd/system/tomcat.service

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment='CATALINA\_OPTS=-Xms512M -Xmx1024M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom'

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

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

User=tomcat

Group=tomcat

UMask=0007

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

Lorsque vous avez terminé, enregistrez et fermez le fichier.

Ensuite, rechargez le démon systemd afin qu’il connaisse notre fichier de service:

sudo systemctl daemon-reload

Démarrez le service Tomcat en tapant:

sudo systemctl start tomcat

Vérifiez que tout a commencé sans erreur en tapant:

sudo systemctl status tomcat

Vous devriez voir une sortie similaire à celle-ci:

Output● tomcat.service - Apache Tomcat Web Application Container

Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor preset: enabled)

Active: active (running) since Thu 2019-10-24 17:18:11 UTC; 4s ago

Process: 5962 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)

Main PID: 5970 (java)

Tasks: 44 (limit: 2377)

Memory: 184.2M

CGroup: /system.slice/tomcat.service

└─5970 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.u

Oct 24 17:18:10 tomcat systemd[1]: Starting Apache Tomcat Web Application Container...

Oct 24 17:18:11 tomcat startup.sh[5962]: Tomcat started.

Oct 24 17:18:11 tomcat systemd[1]: Started Apache Tomcat Web Application Container.

Cela confirme que Tomcat est opérationnel sur votre serveur.

Étape 6 - Ajustez le pare-feu et testez le serveur Tomcat

Maintenant que le service Tomcat est démarré, nous pouvons tester pour nous assurer que la page par défaut est disponible.

Avant de faire cela, nous devons ajuster le pare-feu pour permettre à nos demandes d’accéder au service. Si vous avez suivi les conditions préalables, vous aurez actuellement un pare-feu + ufw + activé.

Tomcat utilise le port + 8080 + pour accepter les demandes. Autorisez le trafic sur ce port en tapant:

sudo ufw allow 8080

Avec le pare-feu modifié, vous pouvez accéder à la page de démarrage par défaut en accédant à votre domaine ou à votre adresse IP, suivis de «+: 8080 +» dans un navigateur Web:

Open in web browserhttp://:8080

Vous verrez la page d’accueil Tomcat par défaut, en plus d’autres informations. Toutefois, si vous cliquez sur les liens de l’application Manager, par exemple, l’accès vous sera refusé. Nous pouvons configurer cet accès ensuite.

Si vous avez réussi à accéder à Tomcat, le moment est venu d’activer le fichier de service de sorte que Tomcat démarre automatiquement au démarrage:

sudo systemctl enable tomcat

Étape 7 - Configurer l’interface de gestion Web Tomcat

Pour utiliser l’application Web de gestion fournie avec Tomcat, nous devons ajouter une connexion à notre serveur Tomcat. Nous allons faire cela en éditant le fichier + tomcat-users.xml:

sudo nano /opt/tomcat/conf/tomcat-users.xml

Vous voudrez ajouter un utilisateur pouvant accéder aux + manager-gui + et + admin-gui + (applications Web fournies avec Tomcat). Vous pouvez le faire en définissant un utilisateur, similaire à l’exemple ci-dessous, entre les balises + tomcat-users +. Assurez-vous de changer le nom d’utilisateur et le mot de passe pour quelque chose de sécurisé:

tomcat-users.xml

<tomcat-users>

. . .

<user username="" password="" roles="manager-gui,admin-gui"/>

</tomcat-users>

Enregistrez et fermez le fichier lorsque vous avez terminé.

Par défaut, les versions les plus récentes de Tomcat limitent l’accès aux applications Manager et Host Manager aux connexions provenant du serveur lui-même. Puisque nous installons sur une machine distante, vous voudrez probablement supprimer ou modifier cette restriction. Pour changer les restrictions d’adresse IP, ouvrez les fichiers + context.xml + appropriés.

Pour l’application Manager, tapez:

sudo nano /opt/tomcat/webapps/manager/META-INF/context.xml

Pour l’application Host Manager, tapez:

sudo nano /opt/tomcat/webapps/host-manager/META-INF/context.xml

À l’intérieur, commentez la restriction d’adresse IP pour autoriser les connexions de n’importe où. Si vous souhaitez uniquement autoriser l’accès aux connexions provenant de votre propre adresse IP, vous pouvez également ajouter votre adresse IP publique à la liste:

Fichiers context.xml pour les applications Web Tomcat

<Context antiResourceLocking="false" privileged="true" >

<Valve className="org.apache.catalina.valves.RemoteAddrValve"

allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" />

</Context>

Enregistrez et fermez les fichiers lorsque vous avez terminé.

Pour appliquer nos modifications, redémarrez le service Tomcat:

sudo systemctl restart tomcat

Étape 8 - Accéder à l’interface Web

Maintenant que nous avons créé un utilisateur, nous pouvons à nouveau accéder à l’interface de gestion Web dans un navigateur Web. Une fois encore, vous pouvez accéder à la bonne interface en entrant le nom de domaine ou l’adresse IP de votre serveur, suivi sur le port 8080 de votre navigateur:

Open in web browserhttp://:8080

La page que vous voyez devrait être la même que celle qui vous avait été fournie lors des tests précédents:

image: https: //assets.digitalocean.com/articles/tomcat9/home-screen.png [racine Tomcat]

Jetons un coup d’œil à l’application Manager, accessible via le lien ou + http: //: 8080 / manager / html +. Vous devrez entrer les informations d’identification du compte que vous avez ajoutées au fichier + tomcat-users.xml +. Ensuite, vous devriez voir une page qui ressemble à ceci:

image: https: //assets.digitalocean.com/articles/tomcat9/manager.png [Gestionnaire d’applications Web Tomcat]

Web Application Manager est utilisé pour gérer vos applications Java. Vous pouvez démarrer, arrêter, recharger, déployer et annuler le déploiement ici. Vous pouvez également exécuter des diagnostics sur vos applications (c’est-à-dire trouver des fuites de mémoire). Enfin, des informations sur votre serveur sont disponibles tout en bas de cette page.

Voyons maintenant le gestionnaire d’hôte, accessible via le lien ou + http: //: 8080 / host-manager / html / +:

image: https: //assets.digitalocean.com/articles/tomcat9/host-manager.png [Gestionnaire d’hôte virtuel Tomcat]

À partir de la page Virtual Host Manager, vous pouvez ajouter des hôtes virtuels pour servir vos applications.

Conclusion

Votre installation de Tomcat est terminée! Vous êtes maintenant libre de déployer vos propres applications Web Java.

Actuellement, votre installation de Tomcat est fonctionnelle, mais entièrement non chiffrée. Cela signifie que toutes les données, y compris les éléments sensibles tels que les mots de passe, sont envoyées en texte brut qui peut être intercepté et lu par d’autres parties sur Internet. Afin d’éviter cela, il est fortement recommandé de chiffrer vos connexions avec SSL. Pour savoir comment chiffrer vos connexions à Tomcat, consultez la page <https://www.digitalocean.com/community/tutorials/how-to-encrypt-tomcat-8-connections-with-apache-or-nginx-on-ubuntu>. -16-04 [ce guide] (*note: ce guide traite du chiffrement Tomcat 8 sous Ubuntu 16.04*).