



Manually Installing Jamf Pro 10.4.0

Red Hat Enterprise Linux, Ubuntu LTS Server, and Windows Server



© copyright 2002-2018 Jamf. All rights reserved.

Jamf has made all efforts to ensure that this guide is accurate.

Jamf
100 Washington Ave S Suite 1100
Minneapolis, MN 55401-2155
(612) 605-6625

Under the copyright laws, this publication may not be copied, in whole or in part, without the written consent of Jamf.

Apache Tomcat and Tomcat are trademarks of the Apache Software Foundation.

Apple, the Apple logo, macOS, and OS X are trademarks of Apple Inc., registered in the United States and other countries.

IOS is a trademark or registered trademark of Cisco in the United States and other countries.

The CASPER SUITE, COMPOSER®, the COMPOSER Logo®, Jamf, the Jamf Logo, JAMF SOFTWARE®, the JAMF SOFTWARE Logo®, RECON®, and the RECON Logo® are registered or common law trademarks of JAMF SOFTWARE, LLC in the U.S. and other countries.

Linux is the registered trademark of Linus Torvalds in the United States and other countries.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Ubuntu is a registered trademark of Canonical Ltd.

All other product and service names mentioned herein are either registered trademarks or trademarks of their respective companies.

Contents

4 Introduction

5 Red Hat Enterprise Linux

5 Requirements

5 Installation and Configuration

19 Ubuntu LTS Server

19 Requirements

19 Installation and Configuration

32 Windows Server

32 Requirements

33 Installation and Configuration

Introduction

Document Deprecation Notice

This *Manually Installing Jamf Pro* guide will be discontinued in an upcoming release of Jamf Pro. Instructions for manually installing Jamf Pro will be included in the *Jamf Pro Installation and Configuration Guide for Linux* and *Jamf Pro Installation and Configuration Guide for Windows*.

This guide explains how to manually install and configure Jamf Pro on the following Linux and Windows platforms:

- [Red Hat Enterprise Linux](#)
- [Ubuntu LTS Server](#)
- [Windows Server](#)

Red Hat Enterprise Linux

Document Deprecation Notice

This *Manually Installing Jamf Pro* guide will be discontinued in an upcoming release of Jamf Pro. Instructions for manually installing Jamf Pro will be included in the *Jamf Pro Installation and Configuration Guide for Linux*.

This section provides a basic set of steps for manually installing and configuring Jamf Pro 10.4.0 on a Red Hat Enterprise Linux (RHEL) server.

Note: The supporting scripts and configuration used for manual installation differ from an installation using the Jamf Pro Installer for Linux. If you do not want to manually install Jamf Pro, you can use the Jamf Pro Installer for Linux. To obtain the installer along with its installation instructions, log in to Jamf Nation, click **Show alternative downloads**, and then click the **Download** button under "Jamf Pro Installer for Linux" on the following page:

<https://www.jamf.com/jamf-nation/my/products>

Requirements

To complete the instructions in this section, you need a computer with a fresh installation of one of the following operating systems:

- Minimum:
 - Red Hat Enterprise Linux (RHEL) 6.4
- Recommended:
 - Red Hat Enterprise Linux (RHEL) 6.9
 - Red Hat Enterprise Linux (RHEL) 7.5

In addition, you need the following:

- Java 1.8
- Apache Tomcat 7 or 8.x.x (Tomcat 8.5.x is recommended)
- The Jamf Pro web app (`ROOT.war`) and database utility (`JSSDatabaseUtil.jar`)
To obtain these items in the Jamf Pro manual install archive, log in to Jamf Nation, click **Show alternative downloads**, then click the **Download** button below "Jamf Pro Manual Installation" on the following page:
<https://www.jamf.com/jamf-nation/my/products>

Installation and Configuration

Follow the step-by-step instructions in this section to install and configure Jamf Pro on RHEL.

Step 1: Install Java and MySQL

Java and MySQL must be installed on the server before you can create the Jamf Pro database and run the Jamf Pro Installer. For instructions, see the [Installing Java and MySQL](#) Knowledge Base article.

Step 2: Create the Jamf Pro Database

Create a MySQL database in which Jamf Pro can store its data and a MySQL user that can access the database.

A default MySQL database name, username, and password are used throughout the instructions in this section. It is recommended, however, that you use a custom username and password that comply with your organization's security requirements. It is also recommended that you do not use the "root" user. A different database name can also be set if desired.

Note: The password must contain uppercase and lowercase letters, at least one number, at least one special character, and be at least eight characters long.

The default MySQL settings in the instructions below are:

- Database name: jamfsoftware
- Username: jamfsoftware
- Password: Jamfsw03!

Note: If you customize any of the MySQL settings, the Database Connection Properties pane will appear the first time you try to access Jamf Pro in a web browser. You will need follow the instructions on this pane and enter your MySQL settings before you can access Jamf Pro.

1. Access the MySQL command-line utility by executing:

```
mysql -u root -p
```

2. Provide the MySQL "root" user password when prompted.
3. At the "mysql" prompt, create a database by executing:

```
CREATE DATABASE jamfsoftware;
```

You can customize the database name by replacing `jamfsoftware` with the desired name as shown in the following example:

```
CREATE DATABASE myDatabaseName;
```

4. Grant permissions to a MySQL user so that it can access the new database with a command similar to the following:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED BY 'Jamfsw03!';
```

You can customize the MySQL username and password by replacing `jamfsoftware` and `Jamfsw03!` with the desired username and password as shown in the following example:

```
GRANT ALL ON myDatabaseName.* TO 'customUsername'@localhost
IDENTIFIED BY 'customPassword'
```

5. Exit the MySQL command-line utility by typing "exit".

Step 3: Install Tomcat

Apache Tomcat is the web application server that runs Jamf Pro.

You will need URLs to download and verify the Tomcat binary distribution that you intend to install. If you have access to a web browser on the RHEL system or if you are remotely connected to an RHEL shell session from your computer, you may want to copy these URLs just before executing the download commands. If not, you may want to copy the URLs into a text document for reference as you type them.

Copy the Necessary URLs

1. In a web browser, open the Tomcat 8 download page:
<https://tomcat.apache.org/download-80.cgi>
2. On the Tomcat 8 download page, navigate to **Tomcat 8.5.x > Binary Distributions > Core**.
3. Right-click the "tar.gz" link, and choose **Copy Link** to copy the URL.
4. Paste the copied URL into a document for reference, or paste it directly into the download command (see below).
5. On the Tomcat 8 download page, right-click the "sha512" link, and choose **Copy Link** to copy the URL.
6. Paste the copied URL into a document for reference, or paste it directly into the download command (see below).

Prepare the System for Tomcat

Note: Tomcat 8.5.30 is used in the commands in this section. When you execute the commands, substitute "8.5.30" with the specific version of Tomcat 8.5.x that you want to install. The most recent version of Tomcat 8.5.x can be downloaded from the following page:

<https://tomcat.apache.org/download-80.cgi>

You may need to install the "wget" utility to execute the commands found in this guide. Install "wget" by executing:

```
sudo yum install wget
```

1. Create a "tomcat" group by executing:

```
sudo groupadd tomcat
```

2. Create a "tomcat" user by executing:

```
sudo useradd -r -g tomcat -d /opt/apache-tomcat-8.5.30 -s /bin/nologin tomcat
```

3. Create a temporary directory for the downloads and change to the directory by executing:

```
mkdir /tmp/tomcat && cd /tmp/tomcat
```

4. Download the Tomcat binary distribution using the tar.gz URL (see above) with a command similar to:

```
wget http://apache.cs.utah.edu/tomcat/tomcat-8/v8.5.30/bin/apache-tomcat-8.5.30.tar.gz
```

5. Download the Tomcat sha512 reference file using the URL (see above) with a command similar to:

```
wget https://www.apache.org/dist/tomcat/tomcat-8/v8.5.30/bin/apache-tomcat-8.5.30.tar.gz.sha512
```

6. Using the filename of the sha512 reference file, verify the binary distribution has not been modified with a command similar to:

```
sha512sum -c apache-tomcat-8.5.30.tar.gz.sha512
```

The result should be something like:

```
apache-tomcat-8.5.30.tar.gz: OK
```

7. If the result of the check indicates that the downloaded binary is "OK", extract the contents of the file with a command similar to:

```
tar -zxvf apache-tomcat-8.5.30.tar.gz
```

8. Move the extracted contents to the desired location with a command similar to:

```
sudo mv apache-tomcat-8.5.30 /opt/
```

9. Ensure the "tomcat" user and group have ownership of the directory and contents with a command similar to:

```
sudo chown -R tomcat:tomcat /opt/apache-tomcat-8.5.30
```

10. Create a symlink to the directory with a command similar to:

```
sudo ln -s /opt/apache-tomcat-8.5.30 /opt/tomcat
```


Note: Installing Tomcat in a directory named with the version number and then symlinking to this directory allows for an easy Tomcat upgrade path later—simply install a newer version of Tomcat in a new directory with its version number in the name and change the symlink to point to the new version.

Step 4: Create a Tomcat Management Service

You will need to know the path to the JRE to provide it for the management service. When installing the JDK, the "jre" folder will typically be found in the JDK folder. The path to Java will vary depending on the platform, OS version, Open JDK vs. Oracle JDK, etc.

Following are different methods that you can use to find the path to Java on your system:

- Reference the `JAVA_HOME` environment variable:

```
$ echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.161-0.b14.e17_4.x86_64
```

- Use alternatives:

```
$ alternatives --config java

There is 1 program that provides 'java'.

  Selection    Command
-----
*+ 1          java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-
1.8.0.161-0.b14.e17_4.x86_64/jre/bin/java)

Enter to keep the current selection[+], or type selection number:
```

- Use `whereis` to find "java" and follow the breadcrumbs:

```
$ whereis java
java: /usr/bin/java
      /usr/lib/java
      /etc/java
      /usr/share/java
      /usr/share/man/man1/java.1.gz

$ ls -la /usr/bin | grep java
lrwxrwxrwx. 1 root root 22 Apr 16 15:04 java -> /etc/alternatives/java

$ ls -la /etc/alternatives | grep /jre/bin/java
lrwxrwxrwx. 1 root root 73 Apr 16 15:04 java ->
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.161-0.b14.e17_4.x86_64/jre/bin/java
```

Note: To accommodate a large number of computers in Jamf Pro, it may be necessary to allocate additional Java Virtual Machine (JVM) memory to Tomcat. If there are other services running on your server, make sure to leave enough memory to accommodate them when configuring the "maximum memory" (`-Xmx`) setting in the Tomcat management service.

Modern Linux Service Management

Modern Linux platforms, such as RHEL 7, are moving to use the "systemctl" utility. This process is recommended for creating a management service, but it is not supported by legacy OS versions.

1. Using your preferred text editor, create a tomcat systemd service file with a command similar to:

```
sudo vi /etc/systemd/system/tomcat.service
```

2. Paste the following into the tomcat.service file:

```
[Unit]
Description=Apache Tomcat 8.5.x Web Application Container
Wants=network.target
After=syslog.target network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/java/default
Environment=JRE_HOME=$JAVA_HOME/jre
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.net.
preferIPv4Stack=true -Xms256M -Xmx512M'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

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

3. Ensure the JAVA_HOME value matches the path to the JDK installed on the system (see above for guidance on finding this path). Adjust the JRE_HOME path, if necessary, and set the JAVA_OPTS memory settings—particularly the "maximum memory" (-Xmx), if desired.
4. Save the tomcat.service file.
5. Restart the systemd daemon by executing:

```
sudo systemctl daemon-reload
```

6. Start the Tomcat service by executing:

```
sudo systemctl start tomcat
```

7. To check the status of Tomcat, execute:

```
systemctl status tomcat
```

8. Enable auto startup of the Tomcat service at boot by executing:

```
sudo systemctl enable tomcat
```

Legacy Linux Service Management

The legacy service management process should only be used for legacy OS versions (RHEL 6).

1. Using your preferred text editor, create a `init.d` service file with a command similar to:

```
sudo vi /etc/init.d/tomcat
```

2. Paste the following two blocks of code into the `/etc/init.d/tomcat` file, adjusting the `JAVA_HOME` path, if necessary, and the `JAVA_OPTS` memory settings—particularly the "maximum memory" (`-Xmx`), if desired:

Part 1

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          tomcat
# Required-Start:    $local_fs $remote_fs $network
# Required-Stop:     $local_fs $remote_fs $network
# Should-Start:      $named
# Should-Stop:       $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Tomcat.
# Description:       Start the Tomcat servlet engine.
### END INIT INFO

TOMCAT_USER=tomcat
TOMCAT_SHUTDOWN_PORT=8005

export JAVA_HOME=/usr/java/default
export JRE_HOME=$JAVA_HOME/jre
export JAVA_OPTS="-Djava.awt.headless=true -Djava.net.preferIPv4Stack=true
-Xms256M -Xmx512M"
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
export CATALINA_HOME="/opt/tomcat"
export CATALINA_PID="$CATALINA_HOME/bin/tomcat.pid"

running_pid() {
    echo `ps aux | grep org.apache.catalina.startup.Bootstrap | grep -v
grep | awk '{ print $2 }'`
}

tomcat_up() {
    echo `netstat -vatpn | grep LISTEN | grep :$TOMCAT_SHUTDOWN_PORT`
}

status() {
    pid=$(running_pid)
    if [ -n "$pid" ]; then
        echo "Tomcat is running with pid: $pid"
        if [ -n "$(tomcat_up)" ]; then
            echo "Tomcat is up."
        else
            echo "Tomcat is not fully started up."
        fi
    else
        echo "Tomcat is not running"
    fi
}
```

Part 2

```
start() {
    pid=$(running_pid)
    if [ -z "$pid" ]; then
        echo "Starting Tomcat..."
        /bin/su -s /bin/bash ${TOMCAT_USER} -c "$CATALINA_HOME/bin/startup.
sh
        sleep 2
    fi
    status
}

stop() {
    if [ -n "$(tomcat_up)" ]; then
        echo "Stopping Tomcat..."
        /bin/su -s /bin/bash ${TOMCAT_USER} -c "$CATALINA_HOME/bin
/shutdown.sh 20"
    fi
    if [ -f "${CATALINA_PID}" ]; then
        pid=$(cat "$CATALINA_PID")
        echo "Tomcat is not completely started up. Killing pid ${pid}..."
        kill -9 $pid
        rm "$CATALINA_PID"
    fi
    status
}

case $1 in
    start|stop|status) $1;;
    restart) stop; start;;
    *) echo "Usage : $0 <start|stop|restart|status>"; exit 1;;
esac

exit 0
```

3. Save the "tomcat" file.
4. Set the permissions on the "tomcat" script by executing:

```
sudo chmod 755 /etc/init.d/tomcat
```

5. Start the Tomcat service by executing:

```
sudo service tomcat start
```

6. To check the status of Tomcat, execute:

```
sudo service tomcat status
```

7. Enable auto startup of the Tomcat service at boot by executing:

```
sudo chkconfig --levels 345 tomcat on
```

Step 5: Configure the Firewall

On systems where the firewall is enabled, the following instructions will assist you in configuring the firewall to allow inbound access on port 8443, which provides access to the Jamf Pro web application.

Note: If the firewall is not enabled, the following commands are not necessary. However, it is highly recommended that you follow industry best practices for production web-accessible systems, including enabling the firewall. For more information about securing your Jamf Pro server, contact your Jamf account representative.

RHEL 7.x

1. Check to see if the firewall is running by executing:

```
systemctl status firewalld
```

2. If the firewall is "active", edit the firewall configuration to allow access to port 8443 by executing the following commands:

```
firewall-cmd --zone=public --add-port=8443/tcp --permanent  
firewall-cmd --reload
```

RHEL 6.x

1. Check to see if the firewall is running by executing:

```
sudo service iptables status
```

2. If the firewall is running, edit the firewall configuration to allow access to port 8443 by executing the following commands:

```
sudo system-config-firewall-tui
```

3. Choose **Other** or **Customize**, and then manually add port 8443 with TCP protocol. The option you choose depends on whether you have a GUI or shell-only interface.

Step 6: Install Jamf Pro

1. (Optional) Create a `/tmp/jamf` directory in which to temporarily store downloads, tools, and scripts for Jamf Pro by executing:

```
mkdir /tmp/jamf
```

2. (Optional) Copy the files from the Jamf Pro manual download archive (`ROOT.war` and `JSSDatabaseUtil.jar`) to the `/tmp/jamf` directory that you just created.

3. Stop the Tomcat service by executing:

```
sudo systemctl stop tomcat
```

or

```
sudo service tomcat stop
```

4. Move the default Tomcat web app out of the way by executing:

```
sudo mv /opt/tomcat/webapps/ROOT /opt/tomcat/webapps/TOMCAT
```

5. Copy the Jamf Pro web app archive to the Tomcat webapps directory by executing a command similar to the following:

```
sudo cp /tmp/jamf/ROOT.war /opt/tomcat/webapps/
```

The Tomcat service should be running and will automatically unpack the `ROOT.war` file into a `ROOT` directory in the `webapps` directory.

6. Assign appropriate access to the `ROOT.war` by executing:

```
sudo chown tomcat:tomcat /opt/tomcat/webapps/ROOT.war
sudo chmod 750 /opt/tomcat/webapps/ROOT.war
```

7. Generate a keystore to enable SSL for Tomcat by executing a command similar to the following with the `keypass`, `storepass`, `dname`, and `validity` values customized for your environment:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keypass
"changeit" -storepass "changeit" -dname "CN=jamf.mycompany.com,
OU=Jamf IT, O=Jamf, L=Minneapolis, ST=MN, C=US"
-keystore /opt/tomcat/keystore -validity <numdays>
```

Make sure to enter the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jamf.mycompany.com
OU=	Organizational unit	Jamf IT
O=	Organization	Jamf
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

8. Back up the Tomcat `server.xml` configuration file by executing:

```
sudo cp /opt/tomcat/conf/server.xml /opt/tomcat/conf/server.xml.bak
```

9. Open the `server.xml` configuration file in a text editor with a command similar to:

```
sudo vi /opt/tomcat/conf/server.xml
```

10. Locate the following comment section in the file (note that this is the Connector on port 8443 not the Connector on port 8443 with HTTP/2 section):

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      This connector uses the NIO implementation.
      ...
-->
```

11. Replace the commented out (initial `<!--` and trailing `-->`) Connector tag immediately following the comment (shown in "a" below) with the Connector tag text shown in "b" below:

a.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      ...
-->
<!--
<Connector port="8443" ...
      ...
/>
-->
```


b.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
...
-->
<Connector port="8443"
    protocol="org.apache.coyote.http11.Http11Nio2Protocol"
    SSLEnabled="true"
    URIEncoding="UTF-8"
    server="Apache"
    executor="tomcatThreadPool"
    maxPostSize="-1"
    scheme="https"
    sslImplementationName="org.apache.tomcat.util.net.jsse.
JSSEImplementation"
    secure="true"
    clientAuth="false"
    sslProtocol="TLS"
    sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1"
    keystoreFile="/opt/tomcat/keystore"
    keystorePass="changeit"
    ciphers="TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
            TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
            TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
            TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
            TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
            TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
            TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
            TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
            TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
            TLS_RSA_WITH_AES_128_CBC_SHA" />
```

12. Before saving the file, ensure the `keystorePass` value is set to the value you used for `storepass` when creating the keystore.
13. Save the `server.xml` file.
14. Start the tomcat service by executing:

```
sudo systemctl start tomcat
```

or

```
sudo service tomcat start
```

15. Verify that Jamf Pro is running by opening a web browser and accessing Jamf Pro over port 8443. The Jamf Pro URL should be something like this: <https://jamf.mycompany.com:8443/>

Note: If you have customized any of the MySQL settings, the Database Connection Properties pane will appear the first time you try to access Jamf Pro in a web browser. You will need to follow the instructions on this pane and enter your MySQL settings before you can access Jamf Pro.

Step 7: Install the Jamf Pro Database Utility

1. Create a directory in which to keep the Jamf Pro Database Utility by executing a command similar to the following:

```
sudo mkdir /opt/jamf
```

2. Copy the Jamf Pro Database Utility to the new directory by executing a command similar to the following:

```
sudo cp /tmp/jamf/JSSDatabaseUtil.jar /opt/jamf/
```

It is highly recommended that you regularly create backups as you work toward a fully configured and operational Jamf Pro. See the [Backing Up the Database](#) Knowledge Base article for instructions on using the database utility to create and schedule backups.

Note: Keep in mind that you have located the database utility at a different path than shown in the article.

Ubuntu LTS Server

Document Deprecation Notice

This *Manually Installing Jamf Pro* guide will be discontinued in an upcoming release of Jamf Pro. Instructions for manually installing Jamf Pro will be included in the *Jamf Pro Installation and Configuration Guide for Linux*.

This section provides a basic set of steps for manually installing and configuring Jamf Pro 10.4.0 on an Ubuntu LTS Server.

Note: The supporting scripts and configuration used for manual installation differ from an installation using the Jamf Pro Installer for Linux. If you do not want to manually install Jamf Pro, you can use the Jamf Pro Installer for Linux. To obtain the installer along with its installation instructions, log in to Jamf Nation, click **Show alternative downloads**, and then click the **Download** button under "Jamf Pro Installer for Linux" on the following page:

<https://www.jamf.com/jamf-nation/my/products>

Requirements

To complete the instructions in this section, you need a computer with a fresh installation of one of the following operating systems:

- Minimum:
 - Ubuntu 12.04 LTS Server (64-bit)
- Recommended:
 - Ubuntu 14.04 LTS Server (64-bit)
 - Ubuntu 16.04 LTS Server (64-bit)

In addition, you need the following:

- Java 1.8
- Apache Tomcat 7 or 8.x.x (Tomcat 8.5.x is recommended)
- The Jamf Pro web app (`ROOT.war`) and database utility (`JSSDatabaseUtil.jar`)
To obtain these items in the Jamf Pro manual install archive, log in to Jamf Nation, click **Show alternative downloads**, and then click the **Download** button below "Jamf Pro Manual Installation" on the following page:

<https://www.jamf.com/jamf-nation/my/products>

Installation and Configuration

Follow the step-by-step instructions in this section to install and configure Jamf Pro on Ubuntu LTS Server.

Step 1: Install Java and MySQL

Java and MySQL must be installed on the server before you can create the Jamf Pro database and run the Jamf Pro Installer. For instructions, see the [Installing Java and MySQL](#) Knowledge Base article.

Step 2: Create the Jamf Pro Database

Create a MySQL database in which Jamf Pro can store its data and a MySQL user that can access the database.

A default MySQL database name, username, and password are used throughout the instructions in this section. It is recommended, however, that you use a custom username and password that comply with your organization's security requirements. It is also recommended that you not use "root" as a username. A different database name can also be set if desired.

The default MySQL settings used in the instructions below are:

- Database name: jamfsoftware
- Username: jamfsoftware
- Password: Jamfsw03!

Note: If you customize any of the MySQL settings, the Database Connection Properties pane will appear the first time you try to access Jamf Pro in a web browser. You will need to follow the instructions on this pane and enter your MySQL settings before you can access Jamf Pro.

1. Access the MySQL command-line utility by executing:

```
mysql -u root -p
```

2. Provide the MySQL "root" user password when prompted.
3. At the "mysql" prompt, create a database by executing:

```
CREATE DATABASE jamfsoftware;
```

You can customize the database name by replacing `jamfsoftware` with the desired name as shown in the following example:

```
CREATE DATABASE myDatabaseName;
```

4. Grant permissions to a MySQL user so that it can access the new database with a command similar to the following:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED BY 'Jamfsw03!';
```

You can customize the MySQL username and password by replacing `jamfsoftware` and `Jamfsw03!` with the desired username and password as shown in the following example:

```
GRANT ALL ON myDatabaseName.* TO 'customUsername'@localhost IDENTIFIED BY  
'customPassword';
```

5. Exit the MySQL command-line utility by typing "exit".

Step 3: Install Tomcat

Apache Tomcat is the web application server that runs Jamf Pro.

You will need URLs to download and verify the Tomcat binary distribution that you intend to install. If you have access to a web browser on the Ubuntu system or if you are remotely connected to an Ubuntu shell session from your computer, you may want to copy these URLs just before executing the download commands. If not, you may want to copy the URLs into a text document for reference as you type them.

Copy the Necessary URLs

1. In a web browser, open the Tomcat 8 download page:
<https://tomcat.apache.org/download-80.cgi>
2. On the Tomcat 8 download page, navigate to **Tomcat 8.5.x > Binary Distributions > Core**.
3. Right-click the "tar.gz" link, and choose **Copy Link** to copy the URL.
4. Paste the copied URL into a document for reference, or paste it directly into the download command (see below).
5. On the Tomcat 8 download page, right-click the "sha512" link, and choose **Copy Link** to copy the URL.
6. Paste the copied URL into a document for reference, or paste it directly into the download command (see below).

Prepare the System for Tomcat

Note: Tomcat 8.5.30 is used in the commands in this section. When you execute the commands, substitute "8.5.30" with the specific version of Tomcat 8.5.x that you want to install. The most recent version of Tomcat 8.5.x can be downloaded from the following page:

<https://tomcat.apache.org/download-80.cgi>

1. Create a "tomcat" group by executing:

```
sudo groupadd tomcat
```

2. Create a "tomcat" user by executing:

```
sudo useradd -r -g tomcat -d /opt/apache-tomcat-8.5.30 -s /bin/nologin  
tomcat
```

3. Create a temporary directory for the downloads and change to the directory by executing:

```
mkdir /tmp/tomcat && cd /tmp/tomcat
```

4. Download the Tomcat binary distribution using the `tar.gz` URL (see above) with a command similar to:

```
wget http://apache.cs.utah.edu/tomcat/tomcat-8/v8.5.30/bin/apache-tomcat-8.5.30.tar.gz
```

5. Download the Tomcat sha512 reference file using the URL (see above) with a command similar to:

```
wget https://www.apache.org/dist/tomcat/tomcat-8/v8.5.30/bin/apache-tomcat-8.5.30.tar.gz.sha512
```

6. Using the filename of the sha512 reference file, verify the binary distribution has not been modified with a command similar to:

```
sha512sum -c apache-tomcat-8.5.30.tar.gz.sha512
```

The result should be something like:

```
apache-tomcat-8.5.30.tar.gz: OK
```

7. If the result of the check indicates that the downloaded binary is "OK", extract the contents of the file with a command similar to:

```
tar -zxvf apache-tomcat-8.5.30.tar.gz
```

8. Move the extracted contents to the desired location with a command similar to:

```
sudo mv apache-tomcat-8.5.30 /opt/
```

9. Ensure the "tomcat" user and group have ownership of the directory and contents with a command similar to:

```
sudo chown -R tomcat:tomcat /opt/apache-tomcat-8.5.30
```

10. Create a symlink to the directory with a command similar to:

```
sudo ln -s /opt/apache-tomcat-8.5.30 /opt/tomcat
```

Note: Installing Tomcat in a directory named with the version number and then symlinking to this directory allows for an easy Tomcat upgrade path later—simply install a newer version of Tomcat in a new directory with its version number in the name and change the symlink to point to the new version.

Step 4: Create a Tomcat Management Service

You will need to know the path to the JRE to provide it for the management service. When installing the JDK, the "jre" folder will typically be found in the JDK folder. The path to Java will vary depending on the platform, OS version, Open JDK vs. Oracle JDK, etc.

Following are different methods that you can use to find the path to Java on your system:

- Reference the `JAVA_HOME` environment variable:

```
$ echo $JAVA_HOME
/usr/lib/jvm/java-8-openjdk-amd64
```

- Use `update-alternatives`:

```
$ sudo update-alternatives --config java
There is only one alternative in link group java (providing /usr/bin
/java): /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
Nothing to configure.
```

- Use `whereis` to find "java" and follow the breadcrumbs:

```
$ whereis java
java: /usr/bin/java
      /usr/share/java
      /usr/lib/jvm/java-8-openjdk-amd64/bin/java
      /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
      /usr/share/man/man1/java.1.gz

$ ls -la /usr/bin | grep java$
lrwxrwxrwx 1 root root 22 Apr 16 18:09 java -> /etc/alternatives/java

$ ls -la /etc/alternatives | grep java$
lrwxrwxrwx 1 root root 46 Apr 16 18:09 java ->
/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java
```

Note: To accommodate a large number of computers in Jamf Pro, it may be necessary to allocate additional Java Virtual Machine (JVM) memory to Tomcat. If there are other services running on your server, make sure to leave enough memory to accommodate them when configuring the "maximum memory" (`-Xmx`) setting in the Tomcat management service.

Modern Linux Service Management

Modern Linux platforms, such as Ubuntu 16, are moving to use the "systemctl" utility. This process is recommended for creating a management service, but it is not supported by legacy OS versions.

1. Using your preferred text editor, create a tomcat systemd service file with a command similar to:

```
sudo vi /etc/systemd/system/tomcat.service
```

2. Paste the following into the `tomcat.service` file:

```
[Unit]
Description=Apache Tomcat 8.5.x Web Application Container
Wants=network.target
After=syslog.target network.target

[Service]
Type=forking

Environment=JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
Environment=JRE_HOME=$JAVA_HOME/jre
Environment=CATALINA_PID=/opt/tomcat/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat
Environment='CATALINA_OPTS=-server -XX:+UseParallelGC'
Environment='JAVA_OPTS=-Djava.awt.headless=true -Djava.net.preferIPv4Stack=true -Xms256M -Xmx512M'

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
SuccessExitStatus=143

User=tomcat
Group=tomcat
UMask=0007
RestartSec=10
Restart=always

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

3. Ensure the `JAVA_HOME` value matches the path to the JDK installed on the system (see above for guidance on finding this path). Adjust the `JRE_HOME` path, if necessary, and set the `JAVA_OPTS` memory settings—particularly the "maximum memory" (`-Xmx`), if desired.
4. Save the `tomcat.service` file.
5. Restart the `systemd` daemon by executing:

```
sudo systemctl daemon-reload
```

6. Start the Tomcat service by executing:

```
sudo systemctl start tomcat
```

7. To check the status of Tomcat, execute:

```
systemctl status tomcat
```

Note: You need to press the Q key to exit from the status reporting.

8. Enable auto startup of the Tomcat service at boot by executing:

```
sudo systemctl enable tomcat
```

Legacy Linux Service Management

The legacy service management process should only be used for legacy OS versions (Ubuntu 14).

1. Using your preferred text editor, create a `init.d` service file with a command similar to:

```
sudo vi /etc/init.d/tomcat
```

2. Paste the following two blocks of code into the `/etc/init.d/tomcat` file, adjusting the `JAVA_HOME` path, if necessary, and the `JAVA_OPTS` memory settings—particularly the "maximum memory" (`-Xmx`), if desired:

Part 1

```
#!/bin/bash
### BEGIN INIT INFO
# Provides:          tomcat
# Required-Start:    $local_fs $remote_fs $network
# Required-Stop:     $local_fs $remote_fs $network
# Should-Start:      $named
# Should-Stop:       $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start Tomcat.
# Description:       Start the Tomcat servlet engine.
### END INIT INFO

TOMCAT_USER=tomcat
TOMCAT_SHUTDOWN_PORT=8005

export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_144
export JRE_HOME=$JAVA_HOME/jre
export JAVA_OPTS="-Djava.awt.headless=true -Djava.net.preferIPv4Stack=true
-Xms256M -Xmx512M"
export PATH=$JAVA_HOME/bin:$JRE_HOME/bin:$PATH
export CATALINA_HOME="/opt/tomcat"
export CATALINA_PID="$CATALINA_HOME/bin/tomcat.pid"

running_pid() {
    echo `ps aux | grep org.apache.catalina.startup.Bootstrap | grep -v
grep | awk '{ print $2 }'`
}

tomcat_up() {
    echo `netstat -vatpn | grep LISTEN | grep :$TOMCAT_SHUTDOWN_PORT`
}
```

Part 2

```
status() {
    pid=$(running_pid)
    if [ -n "$pid" ]; then
        echo "Tomcat is running with pid: $pid"
        if [ -n "$(tomcat_up)" ]; then
            echo "Tomcat is up."
        else
            echo "Tomcat is not fully started up."
        fi
    else
        echo "Tomcat is not running"
    fi
}

start() {
    pid=$(running_pid)
    if [ -z "$pid" ]; then
        echo "Starting Tomcat..."
        /bin/su -s /bin/bash ${TOMCAT_USER} -c $CATALINA_HOME/bin/startup.
sh
        sleep 2
    fi
    status
}

stop() {
    if [ -n "$(tomcat_up)" ]; then
        echo "Stopping Tomcat..."
        /bin/su -s /bin/bash ${TOMCAT_USER} -c "$CATALINA_HOME/bin
/shutdown.sh 20"
    fi
    if [ -f "${CATALINA_PID}" ]; then
        pid=$(cat "$CATALINA_PID")
        echo "Tomcat is not completely started up. Killing pid ${pid}..."
        kill -9 $pid
        rm "$CATALINA_PID"
    fi
    status
}

case $1 in
    start|stop|status) $1;;
    restart) stop; start;;
    *) echo "Usage : $0 <start|stop|restart|status>"; exit 1;;
esac

exit 0
```

3. Save the "tomcat" file.
4. Set the permissions on the "tomcat" script by executing:

```
sudo chmod 755 /etc/init.d/tomcat
```

5. Start the Tomcat service by executing:

```
sudo service tomcat start
```

6. To check the status of Tomcat, execute:

```
sudo service tomcat status
```

7. Enable auto startup of the Tomcat service at boot by executing:

```
sudo update-rc.d tomcat defaults
```

Step 5: Configure the Firewall

On systems where the firewall is enabled, the following instructions will assist you in configuring the firewall to allow inbound access on port 8443, which provides access to the Jamf Pro web application.

Note: If the firewall is not enabled, the following commands are not necessary. However, it is highly recommended that you follow industry best practices for production web-accessible systems, including enabling the firewall. For more information about securing your Jamf Pro server, contact your Jamf account representative.

1. Check to see if the firewall is running by executing:

```
sudo ufw status
```

2. If the firewall is running, edit the firewall configuration to allow access to port 8443 by executing the following commands:

```
sudo ufw allow 8443/tcp  
sudo ufw reload
```

Step 6: Install Jamf Pro

1. (Optional) Create a `/tmp/jamf` directory in which to temporarily store downloads, tools, and scripts for Jamf Pro by executing:

```
mkdir /tmp/jamf
```

2. (Optional) Copy the files from the Jamf Pro manual download archive (`ROOT.war` and `JSSDatabaseUtil.jar`) to the `/tmp/jamf` directory that you just created.

3. Stop the Tomcat service by executing:

```
sudo systemctl stop tomcat
```

or

```
sudo service tomcat stop
```

4. Move the default Tomcat web app out of the way by executing:

```
sudo mv /opt/tomcat/webapps/ROOT /opt/tomcat/webapps/TOMCAT
```

5. Copy the Jamf Pro web app archive to the Tomcat `webapps` directory by executing a command similar to the following:

```
sudo cp /tmp/jamf/ROOT.war /opt/tomcat/webapps/
```

The Tomcat service should be running and will automatically unpack the `ROOT.war` file into a `ROOT` directory in the `webapps` directory.

6. Assign appropriate access to the `ROOT.war` by executing:

```
sudo chown tomcat:tomcat /opt/tomcat/webapps/ROOT.war
sudo chmod 750 /opt/tomcat/webapps/ROOT.war
```

7. Generate a keystore to enable SSL for Tomcat by executing a command similar to the following with the `keypass`, `storepass`, `dname`, and `validity` values customized for your environment:

```
sudo keytool -genkey -alias tomcat -keyalg RSA -keypass
"changeit" -storepass "changeit" -dname "CN=jamf.mycompany.com,
OU=Jamf IT, O=Jamf, L=Minneapolis, ST=MN, C=US"
-keystore /opt/tomcat/keystore -validity <numdays>
```

Make sure to enter the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jamf.mycompany.com
OU=	Organizational unit	Jamf IT
O=	Organization	Jamf
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

8. Back up the Tomcat `server.xml` configuration file by executing:

```
sudo cp /opt/tomcat/conf/server.xml /opt/tomcat/conf/server.xml.bak
```

9. Open the `server.xml` configuration file in a text editor with a command similar to:

```
sudo vi /opt/tomcat/conf/server.xml
```

10. Locate the following comment section in the file (note that this is the Connector on port 8443 not the Connector on port 8443 with HTTP/2 section):

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      This connector uses the NIO implementation.
      ...
-->
```

11. Replace the commented out (initial <!-- and trailing -->) Connector tag immediately following the comment (shown in "a" below) with the Connector tag text shown in "b" below:

a.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      ...
-->
<!--
<Connector port="8443" ...
      ...
</Connector>
-->
```

b.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
...
-->
<Connector port="8443"
    protocol="org.apache.coyote.http11.Http11Nio2Protocol"
    SSLEnabled="true"
    URIEncoding="UTF-8"
    server="Apache"
    executor="tomcatThreadPool"
    maxPostSize="-1"
    scheme="https"
    sslImplementationName="org.apache.tomcat.util.net.jsse.
JSSEImplementation"
    secure="true"
    clientAuth="false"
    sslProtocol="TLS"
    sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1"
    keystoreFile="/opt/tomcat/keystore"
    keystorePass="changeit"
    ciphers="TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
            TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
            TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
            TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
            TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
            TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
            TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
            TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
            TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
            TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
            TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
            TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
            TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
            TLS_RSA_WITH_AES_128_CBC_SHA" />
```

12. Before saving the file, ensure the `keystorePass` value is set to the value you used for `storepass` when creating the keystore.
13. Save the `server.xml` file.
14. Start the tomcat service by executing:

```
sudo systemctl start tomcat
```

or

```
sudo service tomcat start
```

15. Verify that Jamf Pro is running by opening a web browser and accessing Jamf Pro over port 8443. The Jamf Pro URL should be something like this: <https://jamf.mycompany.com:8443/>

Note: If you have customized any of the MySQL settings, the Database Connection Properties pane will appear the first time you try to access Jamf Pro in a web browser. You will need to follow the instructions on this pane and enter your MySQL settings before you can access Jamf Pro.

Step 7: Install the Jamf Pro Database Utility

1. Create a directory in which to keep the Jamf Pro Database Utility by executing a command similar to the following:

```
sudo mkdir /opt/jamf
```

2. Copy the Jamf Pro Database Utility to the new directory by executing a command similar to the following:

```
sudo cp /tmp/jamf/JSSDatabaseUtil.jar /opt/jamf/
```

It is highly recommended that you regularly create backups as you work toward a fully configured and operational Jamf Pro. See the [Backing Up the Database](#) Knowledge Base article for instructions on using the database utility to create and schedule backups.

Note: Keep in mind that you have located the database utility at a different path than shown in the article.

Windows Server

Document Deprecation Notice

This *Manually Installing Jamf Pro* guide will be discontinued in an upcoming release of Jamf Pro. Instructions for manually installing Jamf Pro will be included in the *Jamf Pro Installation and Configuration Guide for Windows*.

This section provides a basic set of steps for manually installing and configuring Jamf Pro 10.4.0 on Windows Server.

Note: The supporting scripts and configuration used for manual installation differ from an installation using the Jamf Pro Installer for Windows. If you do not want to manually install Jamf Pro, you can use the Jamf Pro Installer for Windows. To obtain the installer along with its installation instructions, log in to Jamf Nation, click **Show alternative downloads**, and then click the **Download** button under "Jamf Pro Installer for Windows" on the following page:

<https://www.jamf.com/jamf-nation/my/products>

Requirements

To complete the instructions in this section, you need a computer with a fresh installation of one of the following operating systems:

- Minimum:
 - Windows Server 2008 R2 (64-bit)
 - Windows Server 2012 (64-bit)
- Recommended:
 - Windows Server 2012 R2 (64-bit)
 - Windows Server 2016 (64-bit)

In addition, you need the following:

- The Jamf Pro web app (`ROOT.war`) and database utility (`JSSDatabaseUtil.jar`)
To obtain these items in the Jamf Pro manual install archive, log in to Jamf Nation, click **Show alternative downloads**, and then click the **Download** button below "Jamf Pro Manual Installation" on the following page:
<https://www.jamf.com/jamf-nation/my/products>
- MySQL MSI Installer-Essentials 5.6.x or 5.7.x, available at:
<https://dev.mysql.com/downloads/mysql/>
Note: The latest versions of MySQL may not be compatible with older Windows operating systems. For information on MySQL supported platforms and versions, see the following webpage:
<https://www.mysql.com/support/supportedplatforms/database.html>

- Java SE Development Kit (JDK) 1.8 for Windows x64
You can download the latest JDK from:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files 1.8 (JDK Update 144 and earlier only)
You can download the latest JCE from:
<http://www.oracle.com/technetwork/java/javase/downloads/jce8-download-2133166.html>
- Apache Tomcat 8.5.x Windows Service Installer, available at:
<http://tomcat.apache.org/download-80.cgi>

Installation and Configuration

Follow the step-by-step instructions in this section to install and configure Jamf Pro on Windows Server.

Step 1: Install Java and MySQL

Java and MySQL must be installed on the server before you can create the Jamf Pro database and run the Jamf Pro Installer. For instructions, see the [Installing Java and MySQL](#) Knowledge Base article.

Step 2: Create the Jamf Pro Database

Create a MySQL database in which Jamf Pro can store its data and a MySQL user that can access the database.

A default MySQL database name, username, and password are used throughout the instructions in this section. It is recommended, however, that you use a custom username and password that comply with your organization's security requirements. It is also recommended that you not use "root" as the username. A different database name can also be set if desired.

The default MySQL settings used in the instructions below are:

- Database name: jamfsoftware
- Username: jamfsoftware
- Password: jamfsw03

Note: If you customize any of the MySQL settings, the Database Connection Properties pane will appear the first time you try to access Jamf Pro in a web browser. You will need to follow the instructions on this pane and enter your MySQL settings before you can access Jamf Pro.

1. Open the MySQL Command Line Client by choosing **Start > All Programs > MySQL > MySQL Server 5.x > MySQL Command Line Client** from the menu bar.
2. In the MySQL Command Line Client, enter the root password, and then press the Return key.
3. At the prompt, create a database by executing:

```
CREATE DATABASE jamfsoftware;
```

You can customize the database name by replacing `jamfsoftware` with the desired name as shown in the following example:

```
CREATE DATABASE myDatabaseName ;
```

4. Grant privileges to a MySQL user so that it can access the new database:

```
GRANT ALL ON jamfsoftware.* TO jamfsoftware@localhost IDENTIFIED  
BY 'jamfsw03' ;
```

You can customize the MySQL username and password by replacing `'jamfsoftware'` and `'jamfsw03'` with the desired username and password as shown in the following example:

```
GRANT ALL ON myDatabaseName.* TO 'customUsername'@localhost  
IDENTIFIED BY 'customPassword' ;
```

5. Type `exit` to close Command Prompt.

Step 3: Install Tomcat

Apache Tomcat is the web application server that runs Jamf Pro. The following instructions explain how to manually configure Tomcat to use a self-signed SSL certificate on port 8443.

1. Launch the Apache Tomcat 8.5.x Windows Service Installer.
2. Follow the onscreen instructions, and choose "Minimum" from the **Select the type of install** pop-up menu.
3. Expand the option for **Tomcat** and select the **Service Startup** option to customize the install.
Note: Selecting the **Service Startup** option automatically starts Tomcat when the computer starts.
4. Click **Next** and use the default settings to configure the rest of the installation.

Step 4: Install Jamf Pro

1. (Optional) Create a `\temp\jamf` directory in which to temporarily store downloads, tools, and scripts for Jamf Pro by executing:

```
mkdir /temp/jamf
```

2. (Optional) Copy the files from the Jamf Pro manual download archive (`ROOT.war` and `JSSDatabaseUtil.jar`) to the `/temp/jamf` directory that you just created.
3. Open Command Prompt.

- Rename the ROOT web app directory to something like TOMCAT in Windows Explorer or at the Command Prompt by executing a command similar to the following:

```
move "C:\Program Files\Apache Software Foundation\Tomcat 8.5\
webapps\ROOT" "C:\Program Files\Apache Software Foundation\
Tomcat 8.5\webapps\TOMCAT"
```

- Copy the Jamf Pro ROOT.war web app to the Tomcat webapps directory in Explorer or at the Command Prompt by executing a command similar to the following:

```
copy C:\temp\jamf\ROOT.war "C:\Program Files\Apache
Software Foundation\Tomcat 8.5\webapps\"
```

The Tomcat service should be running and will automatically unpack the ROOT.war file into a ROOT directory in the webapps directory.

- Generate a keystore to enable SSL for Tomcat by executing a command similar to the following:

```
"C:\Program Files\Java\jre1.8.0_162\bin\keytool.exe" -genkey -alias
tomcat -keyalg RSA -keypass "changeit" -storepass "changeit" -dname
"CN=jamf.mycompany.com, OU=Jamf IT, O=Jamf, L=Minneapolis, ST=MN,
C=US" -keystore "C:\Program Files\Apache Software Foundation\Tomcat
8.5\keystore" -validity <numdays>
```

Make sure to change the following attributes as appropriate to your site:

Attribute	Value	Example
CN=	Fully qualified domain name of the server	jamf.mycompany.com
OU=	Organizational unit	Jamf IT
O=	Organization	Jamf
L=	Location (city or office)	Minneapolis
ST=	State, province, or county	MN
C=	Country or region	US

- Back up the Tomcat server.xml configuration file by executing a command similar to the following:

```
copy "C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\server.
xml" "C:\Program Files\Apache Software Foundation\Tomcat 8.5\conf\server.
xml.bak"
```

- Open the server.xml configuration file.

Note: You can do this with Wordpad, as long as the file is saved as a plain text document named server.xml.

9. Locate the following comment section in the file (note that this is the Connector on port 8443 not the Connector on port 8443 with HTTP/2 section):

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      This connector uses the NIO implementation.
      ...
-->
```

10. Replace the commented out (initial <!-- and trailing -->) Connector tag immediately following the comment (shown in "a" below) with the Connector tag text shown in "b" below:

a.

```
<!-- Define a SSL/TLS HTTP/1.1 Connector on port 8443
      ...
-->
<!--
<Connector port="8443" ...
      ...
/>
-->
```

b.

```
<Connector port="8443"
  protocol="org.apache.coyote.http11.Http11Nio2Protocol"
  SSLEnabled="true"
  URIEncoding="UTF-8"
  server="Apache"
  executor="tomcatThreadPool"
  maxPostSize="-1"
  scheme="https"
  sslImplementationName="org.apache.tomcat.util.net.jsse.
JSSEImplementation"
  secure="true"
  clientAuth="false"
  sslProtocol="TLS"
  sslEnabledProtocols="TLSv1.2,TLSv1.1,TLSv1"
  keystoreFile="C:\Program Files\Apache Software
Foundation\Tomcat 8.5\keystore"
  keystorePass="changeit"
  ciphers="TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA,
          TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256,
          TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256,
          TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA,
          TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384,
          TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384,
          TLS_ECDH_RSA_WITH_AES_128_CBC_SHA,
          TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256,
          TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256,
          TLS_ECDH_RSA_WITH_AES_256_CBC_SHA,
          TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384,
          TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384,
          TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
          TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
          TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
          TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
          TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384,
          TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
          TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
          TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
          TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
          TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
          TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384,
          TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
          TLS_RSA_WITH_AES_128_CBC_SHA" />
```

11. Go back to Command Prompt, and add a rule to the firewall configuration to allow access to port 8443 by executing:

```
netsh advfirewall firewall add rule name=Jamf_Secure dir=in
action=allow protocol=TCP localport=8443
```

Step 5: Allocate Additional Memory to Tomcat

The default Java Maximum Memory Pool for Tomcat on Windows is set to 256 MB. This should be increased to at least 512 MB. To accommodate a large number of computers in Jamf Pro, you will have to allocate additional Java Virtual Machine (JVM) memory to Tomcat. If other services are running on your server, make sure to leave enough memory to accommodate them.

1. Open the Tomcat8w.exe application found in the "C:\Program Files\Apache Software Foundation\Tomcat 8.5\bin" path.
2. Click the **Java** tab.
3. Enter the amount of memory you want to allocate in the **Maximum Memory Pool** field.
4. Click **Apply**.
5. Open Command Prompt and restart Tomcat by executing:

```
net stop Tomcat8
net start Tomcat8
```

6. Verify that Jamf Pro is running by opening a web browser and accessing Jamf Pro over port 8443. The Jamf Pro URL (formerly the JSS URL) should be something like this: <https://jamf.mycompany.com:8443/>

Step 6: Install the Jamf Pro Database Utility

1. Create a directory in which to keep the Jamf Pro Database Utility by executing a command similar to the following:

```
mkdir "C:\Program Files\Jamf"
```

2. Copy the Jamf Pro Database Utility to the new directory in Explorer or at the Command Prompt by executing a command similar to the following:

```
copy "C:\temp\jamf\JSSDatabaseUtil.jar" "C:\Program Files\Jamf\"
```

It is highly recommended that you regularly create backups as you work toward a fully configured and operational Jamf Pro. See the [Backing Up the Database](#) Knowledge Base article for guidance on using the database utility to create and schedule backups.

Note: Keep in mind that you have located the database utility at a different path than shown in the article.