



I've installed Tomcat 9 on Ubuntu 20.04. When I do: `$ sudo systemctl status tomcat ...` I see the following output: ● tomcat.service - Apache Tomcat Web Application Container Loaded: loaded (/etc/systemd/system/tomcat.service; disabled; vendor> Active: activating (auto-restart) since Mon 2026-01-26 20:04:35 EET;> Process: 497561 ExecStart=/home/mperdikeas/software/tomcat/bin/startu> Process: 497578 ExecStop=/home/mperdikeas/software/tomcat/bin/shutdown> Main PID: 497561 (code=exited, status=0/SUCCESS) However, when I go to `http://localhost:8080` I get "This site can't be reached".

The issue is that Tomcat is in a continuous restart loop and isn't properly running. Here are the steps to diagnose and fix this:

## 1. First, check Tomcat's logs for the actual error:

```
bash
```

```
sudo journalctl -u tomcat -n 50 --no-pager
```

Or check Tomcat's own logs:

```
bash
```

```
tail -f /home/mperdikeas/software/tomcat/logs/catalina.out
```

## 2. Common issues and fixes:

### A. Java/JDK not installed or configured:

```
bash
```

```
# Check if Java is installed
java -version
```

```
# If not installed, install OpenJDK:
sudo apt update
sudo apt install openjdk-11-jdk
```

### B. Permissions issue with Tomcat directory:

```
bash
```

```
# Set proper ownership (adjust username as needed)
sudo chown -R mperdikeas:mperdikeas /home/mperdikeas/software/tomcat
sudo chmod +x /home/mperdikeas/software/tomcat/bin/*.sh
```

### C. Environment variables not set in systemd service:

Check your `/etc/systemd/system/tomcat.service` file. It should include Java environment variables:

```
ini
```

```
[Service]
Type=forking
Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64"
Environment="CATALINA_HOME=/home/mperdikeas/software/tomcat"
Environment="CATALINA_BASE=/home/mperdikeas/software/tomcat"
User=mperdikeas
Group=mperdikeas
ExecStart=/home/mperdikeas/software/tomcat/bin/startup.sh
ExecStop=/home/mperdikeas/software/tomcat/bin/shutdown.sh
Restart=on-failure
```

### D. Port 8080 might be in use:

```
bash
```

```
# Check if port 8080 is already occupied
sudo netstat -tlnp | grep :8080
```

## 3. Immediate troubleshooting steps:

Stop the service and start Tomcat manually to see errors:

```
bash
```

```
sudo systemctl stop tomcat
cd /home/mperdikeas/software/tomcat/bin
./startup.sh
```

Check if Tomcat is running:

```
bash
ps aux | grep tomcat
```

Check Tomcat's console output:

```
bash
tail -f /home/mperdikeas/software/tomcat/logs/catalina.out
```

## 4. If manually starting works but systemd doesn't:

- The systemd service file might have incorrect paths or environment variables
- The service is running as a different user without proper permissions
- Check SELinux/AppArmor restrictions (though less common on Ubuntu)

Share the output of these commands for more specific help:

```
bash
java -version
ls -la /home/mperdikeas/software/tomcat/bin/*.sh
cat /etc/systemd/system/tomcat.service
sudo journalctl -u tomcat -n 30
```

This should help identify why Tomcat keeps restarting and not serving on port 8080.



Can you explain the following curiosity: \$ sudo ls /home/mperdikeas/software/tomcat/bin/ bootstrap.jar configtest.sh shutdown.sh catalina.bat daemon.sh startup.bat catalina.sh digest.bat startup.sh catalina-tasks.xml digest.sh tomcat-juli.jar ciphers.bat makebase.bat tomcat-native.tar.gz ciphers.sh makebase.sh tool-wrapper.bat commons-daemon.jar setclasspath.bat tool-wrapper.sh commons-daemon-native.tar.gz setclasspath.sh version.bat configtest.bat shutdown.bat version.sh mperdikeas@mp-ThinkStation-P320:~/software/tomcat# \$ sudo ls /home/mperdikeas/software/tomcat/bin/\*.sh ls: cannot access '/home/mperdikeas/software/tomcat/bin/\*.sh': No such file or directory

This is a classic **shell glob expansion issue** that occurs because you're using `sudo`! Here's what's happening:

## The Problem:

When you run:

```
bash
sudo ls /home/mperdikeas/software/tomcat/bin/*.sh
```

The shell (bash) expands the wildcard `*.sh` **BEFORE** the `sudo` command runs. So it's actually trying to run:

```
bash
sudo ls file1.sh file2.sh file3.sh ... # etc.
```

But when you're not root, you might not have permission to list files in `/home/mperdikeas/software/tomcat/bin/` (since `mperdikeas` owns them), so the wildcard expansion fails and `*.sh` remains literally `*.sh`.

## Solutions:

### Option 1: Run the shell command as root

```
bash
sudo bash -c "ls /home/mperdikeas/software/tomcat/bin/*.sh"
```

### Option 2: Use `sudo` with the `-H` flag (preserves user's environment)

```
bash
sudo -H ls /home/mperdikeas/software/tomcat/bin/*.sh
```

### Option 3: Let `sudo` spawn a new shell

```
bash
sudo sh -c 'ls /home/mperdikeas/software/tomcat/bin/*.sh'
```

#### Option 4: First become root, then run the command

```
bash
sudo -i
ls /home/mperdikeas/software/tomcat/bin/*.sh
exit
```

#### Option 5: Use find instead (doesn't rely on shell expansion)

```
bash
sudo find /home/mperdikeas/software/tomcat/bin/ -name "*.sh" -exec ls -la {} \;
```

### Why the first command works:

```
bash
sudo ls /home/mperdikeas/software/tomcat/bin/
```

This works because there's no wildcard expansion needed - you're just listing the directory contents, which sudo has permission to do.

### Check the actual .sh files exist:

You can verify the .sh files exist with:

```
bash
ls -la /home/mperdikeas/software/tomcat/bin/ | grep "\.sh$"
```

Or as the tomcat user (probably mperdikeas):

```
bash
sudo -u mperdikeas ls /home/mperdikeas/software/tomcat/bin/*.sh
```

This permission issue is likely also affecting your Tomcat startup! When systemd tries to run the .sh scripts, it might be failing due to permission issues. The fact that you need sudo to list the directory suggests the permissions aren't set correctly for the tomcat user to execute these scripts.



So the thing is the following: `$ sudo systemctl start tomcat ...` exits with exit code zero (0). Can I view some logs for systemctl to understand why tomcat can't start properly?

Yes, absolutely! Here are several ways to view logs for systemctl and understand why Tomcat isn't starting properly:

## 1. Journalctl (Most comprehensive)

```
bash
# View the last 50 log entries for tomcat service
sudo journalctl -u tomcat -n 50

# Follow logs in real-time
sudo journalctl -u tomcat -f

# View logs since last boot
sudo journalctl -u tomcat --since boot

# View all logs with full details
sudo journalctl -u tomcat --no-pager

# Show logs with timestamps
sudo journalctl -u tomcat -o short-precise
```

## 2. Systemctl status with more details

```
bash
# More detailed output
sudo systemctl status tomcat -l
```

```
# Show recent journal entries
sudo systemctl status tomcat --no-pager
```

### 3. Check specific failure reasons

```
bash

# Show only failed units
sudo systemctl --failed

# Check if tomcat has any dependency issues
sudo systemctl list-dependencies tomcat

# Check the service's environment
sudo systemctl show tomcat
```

### 4. Increase verbosity in journalctl

```
bash

# Show all fields including error codes
sudo journalctl -u tomcat -o verbose

# Filter only for error priority messages
sudo journalctl -u tomcat -p err
```

### 5. Check Tomcat's own logs directly

```
bash

# Catalina log (main log)
sudo tail -f /home/mperdikeas/software/tomcat/logs/catalina.out

# Check if catalina.out exists or if logs go elsewhere
sudo ls -la /home/mperdikeas/software/tomcat/logs/

# Localhost log
sudo tail -f /home/mperdikeas/software/tomcat/logs/localhost*.log
```

### 6. Debug systemd service execution

```
bash

# Run the service in debug mode (temporarily)
sudo systemctl edit tomcat --full
# Add to [Service] section: StandardOutput=console
# Then: sudo systemctl daemon-reload && sudo systemctl start tomcat

# Or check the actual ExecStart command manually
sudo -u mperdikeas /home/mperdikeas/software/tomcat/bin/startup.sh
```

### 7. Check system logs for any related errors

```
bash

# General system logs
sudo dmesg | tail -20

# Auth logs (for permission issues)
sudo tail -f /var/log/auth.log
```

### Most likely causes based on your situation:

Since `systemctl start tomcat` exits with code 0 but the service stays in "activating (auto-restart)", check for:

1. **Missing JAVA\_HOME** in systemd service file
2. **Permission issues** on Tomcat directories
3. **Port conflict** (8080 already in use)
4. **Incorrect user/group** in service file
5. **Missing environment variables**

**Try this first to see the actual error:**

```
bash

sudo journalctl -u tomcat --since "5 minutes ago" --no-pager
```

If that doesn't show clear errors, run Tomcat manually to see what happens:

```
bash
```

```
cd /home/mperdikeas/software/tomcat/bin  
sudo -u mperdikeas ./startup.sh
```

Share the output from `journalctl` and we can identify the exact issue!

**DEMO WATERMARK**





I see the following: Jan 26 20:23:22 mp-ThinkStation-P320 systemd[1]: Started Apache Tomcat Web Application Container. Jan 26 20:23:22 mp-ThinkStation-P320 startup.sh[505391]: Tomcat started. Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: NOTE: Picked up JDK\_JAVA\_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.lang.invoke=ALL-UNNAMED --add-opens=java.base/java.lang.reflect=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: Jan 26, 2026 8:23:22 PM org.apache.catalina.startup.Catalina stopServer Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: SEVERE: Could not contact [localhost:8005] (use port [8005] and offset [0]). Tomcat may not be running. Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: Jan 26, 2026 8:23:22 PM org.apache.catalina.startup.Catalina stopServer Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: SEVERE: Error stopping Catalina Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: java.net.ConnectException: Connection refused Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/sun.nio.ch.Net.connect0(Native Method) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/sun.nio.ch.Net.connect(Net.java:579) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/sun.nio.ch.Net.connect(Net.java:568) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/sun.nio.ch.NioSocketImpl.connect(NioSocketImpl.java:593) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.net.SocksSocketImpl.connect(SocksSocketImpl.java:327) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.net.Socket.connect(Socket.java:633) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.net.Socket.connect(Socket.java:583) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.net.Socket.<init>(Socket.java:507) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.net.Socket.<init>(Socket.java:287) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at org.apache.catalina.startup.Catalina.stopServer(Catalina.java:630) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:77) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at java.base/java.lang.reflect.Method.invoke(Method.java:569) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at org.apache.catalina.startup.Bootstrap.stopServer(Bootstrap.java:390) Jan 26 20:23:22 mp-ThinkStation-P320 shutdown.sh[505415]: at org.apache.catalina.startup.Bootstrap.main(Bootstrap.java:479) Jan 26 20:23:22 mp-ThinkStation-P320 systemd[1]: tomcat.service: Succeeded.

### Ah! Now I see the problem clearly!

Tomcat is starting and then immediately shutting down. The logs show:

1. startup.sh runs and says "Tomcat started"
2. shutdown.sh immediately runs and tries to stop Tomcat
3. But Tomcat hasn't fully started yet, so shutdown fails with "Connection refused" on port 8005
4. Systemd sees the service exited, so it tries to restart it (auto-restart loop)

## The Root Cause:

Your tomcat.service file is likely configured with **Type=forking** but Tomcat isn't forking properly, or the PID file isn't being created/read correctly. Systemd thinks the service didn't start, so it immediately calls the shutdown script.

## Solutions:

### Option 1: Change systemd service Type to simple

Edit your service file:

```
bash
```

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

Change:

```
ini
```

```
[Service]
Type=simple # instead of forking
ExecStart=/home/mperdikeas/software/tomcat/bin/catalina.sh run
ExecStop=/home/mperdikeas/software/tomcat/bin/shutdown.sh
# Remove PIDFile if it exists
```

### Option 2: Use daemon.sh instead of startup.sh

### Option 2: Use daemon.sh instead of start/cup.sh

Tomcat has a daemon wrapper:

ini

```
[Service]
Type=forking
ExecStart=/home/mperdikeas/software/tomcat/bin/daemon.sh start
ExecStop=/home/mperdikeas/software/tomcat/bin/daemon.sh stop
```

### Option 3: Check and fix the current forking configuration

If you want to keep Type=forking:

ini

```
[Service]
Type=forking
# Add these lines:
PIDFile=/home/mperdikeas/software/tomcat/temp/tomcat.pid
Environment="CATALINA_PID=/home/mperdikeas/software/tomcat/temp/tomcat.pid"
ExecStart=/home/mperdikeas/software/tomcat/bin/startup.sh
ExecStop=/home/mperdikeas/software/tomcat/bin/shutdown.sh
ExecStartPre=/usr/lib/jvm/java-11-openjdk-amd64/bin/java -Xmx1g -Xms1g -Djava.awt.headless=true -Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=9090 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Djava.rmi.server.hostname=localhost -Djdk.management.enabled=1 -jar /home/mperdikeas/software/tomcat/bin/catalina.jar run
```

### Option 4: The simplest fix - use catalina.sh run

ini

```
[Service]
Type=simple
User=mperdikeas
Group=mperdikeas
Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64"
Environment="CATALINA_HOME=/home/mperdikeas/software/tomcat"
ExecStart=/home/mperdikeas/software/tomcat/bin/catalina.sh run
ExecStop=/home/mperdikeas/software/tomcat/bin/shutdown.sh
Restart=on-failure
```

### Immediate test:

First, stop the service and test manually:

bash

```
sudo systemctl stop tomcat
```

```
# Test if Tomcat works with catalina.sh run
cd /home/mperdikeas/software/tomcat/bin
./catalina.sh run
```

If that works (you should see Tomcat startup logs in terminal), then Option 4 above is your solution.

### After making changes:

bash

```
sudo systemctl daemon-reload
sudo systemctl restart tomcat
sudo systemctl status tomcat
```

**Which option would you like to try first?** I recommend Option 4 as it's the most straightforward for debugging and usually works well with systemd.