**Java and JVM Practical Task**

**Install openjdk**

sudo apt update && sudo apt upgrade -y

sudo apt install openjdk-11-jdk -y

java -version

cd /opt

**Download/install Tomcat server.**

sudo wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.97/bin/apache-tomcat-9.0.97.tar.gz

sudo tar xvf apache-tomcat-9.0.97.tar.gz

sudo mv apache-tomcat-9.0.97 tomcat

sudo chown -R ubuntu:ubuntu /opt/tomcat

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

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/java-11-openjdk-amd64

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

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment=CATALINA\_TMPDIR=/opt/tomcat/temp

Environment=JAVA\_OPTS=-Djava.awt.headless=true -Xms512M -Xmx1024M

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

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

[Install]

WantedBy=multi-user.target

sudo systemctl daemon-reload

sudo systemctl enable tomcat

sudo systemctl start tomcat

sudo systemctl status tomcat

http://<ec2-public-ip>:8080

**Enable JMX in Tomcat**

**Create setenv.sh**

cd /opt/tomcat/bin

sudo touch setenv.sh

sudo chmod +x setenv.sh

**Add JMX Configuration to setenv.sh**:

sudo vi /opt/tomcat/bin/setenv.sh

# Enable JMX Remote Access

export CATALINA\_OPTS="$CATALINA\_OPTS -Dcom.sun.management.jmxremote"

export CATALINA\_OPTS="$CATALINA\_OPTS -Dcom.sun.management.jmxremote.port=1099"

export CATALINA\_OPTS="$CATALINA\_OPTS -Dcom.sun.management.jmxremote.ssl=false"

export CATALINA\_OPTS="$CATALINA\_OPTS -Dcom.sun.management.jmxremote.authenticate=false"

-Dcom.sun.management.jmxremote: Enables JMX remote monitoring.

-Dcom.sun.management.jmxremote.port=1099: Specifies the port on which JMX will listen (you can use any available port).

-Dcom.sun.management.jmxremote.ssl=false: Disables SSL (optional but recommended for development; enable for production).

-Dcom.sun.management.jmxremote.authenticate=false: Disables authentication (optional, but again, use true for production environments).

Need to ensure that your EC2 instance's security group allows incoming traffic on the port used by JMX (default is 1099).

Restart Tomcat to apply the changes.

### 

### **1. Ports in Use by the Java Process**

The output of lsof shows which ports are being used by the Java process:

sudo lsof -i -P -n | grep java

java 13915 root 15u IPv6 29635 0t0 TCP \*:33463 (LISTEN)

java 13915 root 16u IPv6 29636 0t0 TCP \*:1099 (LISTEN)

java 13915 root 17u IPv6 29637 0t0 TCP \*:43007 (LISTEN)

java 13915 root 51u IPv6 29638 0t0 TCP \*:8080 (LISTEN)

java 13915 root 57u IPv6 29644 0t0 TCP 127.0.0.1:8005 (LISTEN)

### **Key Ports in Use:**

* **Port 8080**: This is the default **HTTP connector** port for Tomcat, used to serve web applications over HTTP.
* **Port 1099**: This is the port Tomcat is using for **JMX (Java Management Extensions)**. This is where JMX clients (e.g., JConsole, VisualVM) can connect to monitor and manage Tomcat remotely.
* **Port 8005**: This is the **shutdown port** for Tomcat. It’s used for graceful shutdown of Tomcat.
* **Port 33463** and **Port 43007**: These appear to be **random dynamic ports** that are open and listening. These may be used for internal communication or specific application purposes.

**Verify JMX is Enabled**

sudo ss -tuln | grep 1099

**Change CATALINA\_OPTS to use same for RMI as for JMX**

export CATALINA\_OPTS="$CATALINA\_OPTS \

-Dcom.sun.management.jmxremote \

-Dcom.sun.management.jmxremote.port=1099 \

-Dcom.sun.management.jmxremote.ssl=false \

-Dcom.sun.management.jmxremote.authenticate=false \

-Djava.rmi.server.hostname=54.175.199.29 \

-Djava.rmi.server.useCodebaseOnly=false"

**Rerun tomcat with min heap size 10M and max heap size 20M.**

export CATALINA\_OPTS="$CATALINA\_OPTS \

-Xms10m \

-Xmx20m \

**Increase min heap size to 1G and max heap size to 3G, enable parallel garbage collector.**

export CATALINA\_OPTS="$CATALINA\_OPTS \

-Xms1g \

-Xmx3g \

-XX:+UseParallelGC \

**Launch Jenkins WAR as a standalone application, verify that it works.**

java -jar jenkins.war