TomEE and WebSphere MQ

Steps to integrate TomEE with Websphere MQ

1. Unzip rar file place jars under tomee/lib
2. Added the below to conf/tomee.xml

```
    <tomee>

    <Container id="wmq" type="MESSAGE">

    ResourceAdapter=wmqRA

    MessageListenerInterface=javax.jms.MessageListener

    ActivationSpecClass=com.ibm.mq.connector.inbound.ActivationSpecImpl

    </Container>




   <Resource id="wmqRA" type="com.ibm.mq.connector.ResourceAdapterImpl" class-
name="com.ibm.mq.connector.ResourceAdapterImpl">

    connectionConcurrency=5

    maxConnections=10

    logWriterEnabled=true

    reconnectionRetryCount=5

    reconnectionRetryInterval=300000

    traceEnabled=false

    traceLevel=3

   </Resource>



   <Resource **id="qcf"**  type="javax.jms.ConnectionFactory" class-
name="com.ibm.mq.connector.outbound.ManagedConnectionFactoryImpl">

    TransactionSupport=none

    ResourceAdapter=wmqRA

    HostName=10.a.b.c
    Port=1414

    QueueManager=QM_TIERL

   Channel=SYSTEM.ADMIN.SVRCONN

   TransportType=Client

   UserName=xyz

   Password=*****

  </Resource>



  <Resource id="wmq-javax.jms.QueueConnectionFactory"
type="javax.jms.QueueConnectionFactory" class-
name="com.ibm.mq.connector.outbound.ManagedQueueConnectionFactoryImpl">

    TransactionSupport=xa

    ResourceAdapter=wmqRA

  </Resource>



  <Resource id="wmq-javax.jms.TopicConnectionFactory"
type="javax.jms.TopicConnectionFactory" class-
name="com.ibm.mq.connector.outbound.ManagedTopicConnectionFactoryImpl">

    TransactionSupport=xa

    ResourceAdapter=wmqRA

  </Resource>



  <Resource **id="queue"** type="javax.jms.Queue"

class-name="com.ibm.mq.connector.outbound.MQQueueProxy">

    arbitraryProperties

    baseQueueManagerName

    baseQueueName

    CCSID=1208
    encoding=NATIVE
    expiry=APP
    failIfQuiesce=true
    persistence=APP
    priority=APP
    readAheadClosePolicy=ALL
    targetClient=JMS
  </Resource>

  <Resource id="wmq-javax.jms.Topic" type="javax.jms.Topic" class-name="com.ibm.mq.connector.outbound.MQTopicProxy">
    arbitraryProperties
    baseTopicName
    brokerCCDurSubQueue=SYSTEM.JMS.D.CC.SUBSCRIBER.QUEUE
    brokerDurSubQueue=SYSTEM.JMS.D.SUBSCRIBER.QUEUE
    brokerPubQueue
    brokerPubQueueManager
    brokerVersion=1
    CCSID=1208
    encoding=NATIVE
    expiry=APP
    failIfQuiesce=true
    persistence=APP
    priority=APP
    readAheadClosePolicy=ALL
    targetClient=JMS
  </Resource>



 </tomee>



3. In web.xml add the below to access resources

 <resource-ref>

     <res-ref-name>myqcf< /res-ref-name>

    <res-type>javax.jms.ConnectionFactory < /res-type>

    <res-auth>Container</res-auth>< /br>

    <res-sharing-scope>Shareable< /res-sharing-scope>

    <mapped-name>qcf< /mapped-name>

  </resource-ref>



 <resource-env-ref>

   <resource-env-ref-name>myqueue< /resource-env-ref-name>

   <resource-env-ref-type>javax.jms.Queue< /resource-env-ref-type>

   <mapped-name>queue< /mapped-name>

  </resource-env-ref>
```

Code:

```


    @Resource(name = "qcf")

    private ConnectionFactory connectionFactory;

    @Resource(name = "queue")

    private Queue queue;

    Connection connection = connectionFactory.createConnection();

    Session session = connection.createSession(false,
QueueSession.AUTO_ACKNOWLEDGE);
```

```
    MessageProducer producer = session.createProducer(queue);

    TextMessage message = session.createTextMessage();

    message.setText("Test Message");

    connection.start();

    producer.send(message);

    session.close();

    connection.close();
```

Install Docker on Ubuntu

Step 1 — Installing Docker

The Docker installation package available in the official Ubuntu repository may not be the latest version. To ensure we get the latest version, we'll install Docker from the official Docker repository. To do that, we'll add a new package source, add the GPG key from Docker to ensure the downloads are valid, and then install the package.

First, update your existing list of packages:

```
sudo apt update
```

Next, install a few prerequisite packages which let `apt` use packages over HTTPS:

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

Then add the GPG key for the official Docker repository to your system:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

Add the Docker repository to APT sources:

```
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable"
```

This will also update our package database with the Docker packages from the newly added repo.

Make sure you are about to install from the Docker repo instead of the default Ubuntu repo:

```
apt-cache policy docker-ce
```

You’ll see output like this, although the version number for Docker may be different:

Output of apt-cache policy docker-ce

```
docker-ce:
  Installed: (none)
  Candidate: 5:19.03.9~3-0~ubuntu-focal
  Version table:
     5:19.03.9~3-0~ubuntu-focal 500
        500 https://download.docker.com/linux/ubuntu focal/stable amd64
Packages
```

Notice that `docker-ce` is not installed, but the candidate for installation is from the Docker repository for Ubuntu 20.04 (`focal`).

Finally, install Docker:

```
sudo apt install docker-ce
```

Docker should now be installed, the daemon started, and the process enabled to start on boot. Check that it’s running:

```
sudo systemctl status docker
```

The output should be similar to the following, showing that the service is active and running:

```
Output
● docker.service - Docker Application Container Engine
     Loaded: loaded (/lib/systemd/system/docker.service; enabled;
vendor preset: enabled)
     Active: active (running) since Tue 2020-05-19 17:00:41 UTC; 17s
ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 24321 (dockerd)
      Tasks: 8
     Memory: 46.4M
     CGroup: /system.slice/docker.service
             └─24321 /usr/bin/dockerd -H fd:// --
containerd=/run/containerd/containerd.sock
```

Installing Docker now gives you not just the Docker service (daemon) but also the `docker` command line utility, or the Docker client.

Step 2 — Executing the Docker Command Without Sudo (Optional)

By default, the `docker` command can only be run the **root** user or by a user in the **docker** group, which is automatically created during Docker’s installation process. If you attempt to run the `docker` command without prefixing it with `sudo` or without being in the **docker** group, you’ll get an output like this:

```
Output
docker: Cannot connect to the Docker daemon. Is the docker daemon
running on this host?.
See 'docker run --help'.
```

If you want to avoid typing `sudo` whenever you run the `docker` command, add your username to the `docker` group:

```
sudo usermod -aG docker ${USER}
```

To apply the new group membership, log out of the server and back in, or type the following:

```
su - ${USER}
```

You will be prompted to enter your user’s password to continue.

Confirm that your user is now added to the **docker** group by typing:

```
groups
```

```
Output
sammy sudo docker
```

If you need to add a user to the `docker` group that you’re not logged in as, declare that username explicitly using:

```
sudo usermod -aG docker username
```

Step 3 — Using the Docker Command

Using `docker` consists of passing it a chain of options and commands followed by arguments. The syntax takes this form:

```
docker [option] [command] [arguments]
```

To view all available subcommands, type:

```
docker
```

As of Docker 19, the complete list of available subcommands includes:

```
Output
  attach      Attach local standard input, output, and error streams to
a running container
  build       Build an image from a Dockerfile
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local
filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's
filesystem
  events      Get real time events from the server
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  history     Show the history of an image
  images      List images
  import      Import the contents from a tarball to create a filesystem
image
  info        Display system-wide information
  inspect     Return low-level information on Docker objects
  kill        Kill one or more running containers
  load        Load an image from a tar archive or STDIN
  login       Log in to a Docker registry
  logout      Log out from a Docker registry
  logs        Fetch the logs of a container
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the
container
  ps          List containers
  pull        Pull an image or a repository from a registry
  push        Push an image or a repository to a registry
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  rmi         Remove one or more images
  run         Run a command in a new container
  save        Save one or more images to a tar archive (streamed to
STDOUT by default)
  search      Search the Docker Hub for images
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage
statistics
  stop        Stop one or more running containers
  tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  version     Show the Docker version information
  wait        Block until one or more containers stop, then print their
exit codes
```

To view the options available to a specific command, type:

```
docker docker-subcommand --help
```

To view system-wide information about Docker, use:

```
docker info
```

JMS Configuration in Tomcat with IBM MQ

Tomcat is Servlet container which support Servlets and JSPs. Lets connect Tomcat with Messaging provider IBM MQ9. In this we use MQ running in Docker(icr.io/ibm-messaging/mq) and ports binded to host system. Below are MQ objects.

Queue Manager : TomcatQM

Listener : SYSTEM.LISTENER.TCP.1

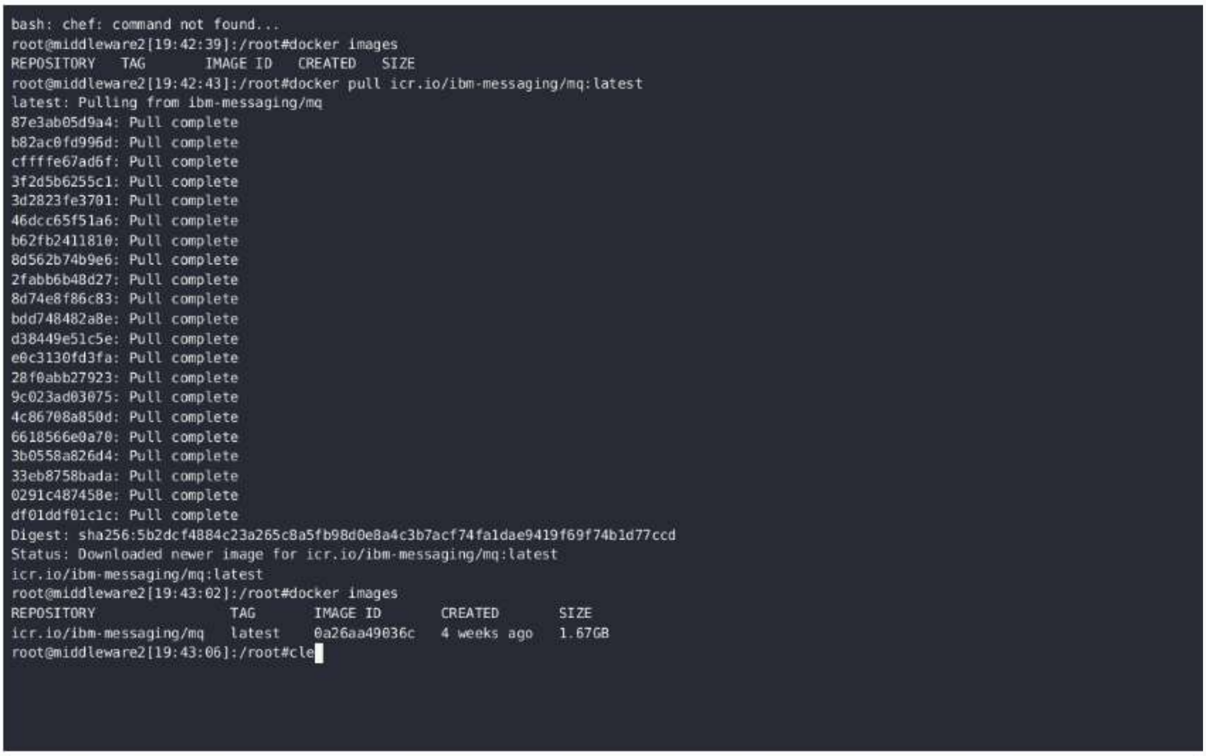
Channel : DEV.APP.SVRCONN

Queue: DEV.QUEUE.1

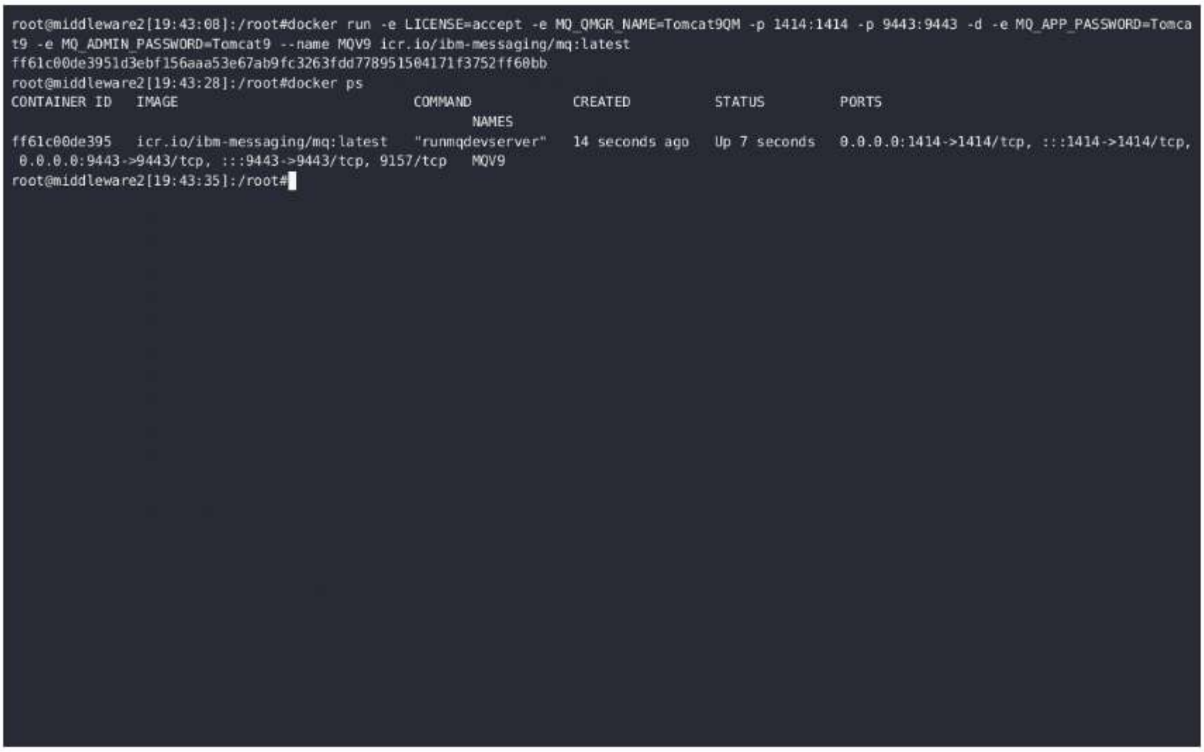
1. Get the MQ Docker image, start the container and check the logs with below commands

```
docker pull icr.io/ibm-messaging/mq:latest
docker run -e LICENSE=accept -e MQ_QMGR_NAME=TomcatQM -p 1414:1414 -p
9443:9443 -d -e MQ_APP_PASSWORD=Tomcat -e MQ_ADMIN_PASSWORD=Tomcat --name
MQV9 icr.io/ibm-messaging/mq:latest
```

```
docker ps
```



```
bash: chef: command not found...
root@middleware2[19:42:39]:/root#docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
root@middleware2[19:42:43]:/root#docker pull icr.io/ibm-messaging/mq:latest
latest: Pulling from ibm-messaging/mq
87e3ab05d9a4: Pull complete
b82ac0fd996d: Pull complete
cffffe67ad6f: Pull complete
3f2d5b6255c1: Pull complete
3d2823fe3701: Pull complete
46dcc65f51a6: Pull complete
b62fb2411810: Pull complete
8d562b74b9e6: Pull complete
2fabb6b48d27: Pull complete
8d74e8f86c83: Pull complete
bdd748482a8e: Pull complete
d38449e51c5e: Pull complete
e0c3130fd3fa: Pull complete
28f0abb27923: Pull complete
9c023ad03075: Pull complete
4c86708a850d: Pull complete
6618566e0a70: Pull complete
3b0558a826d4: Pull complete
33eb8758bada: Pull complete
0291c487458e: Pull complete
df01ddf01c1c: Pull complete
Digest: sha256:5b2dcf4884c23a265c8a5fb98d0e8a4c3b7acf74fa1dae9419f69f74b1d77ccd
Status: Downloaded newer image for icr.io/ibm-messaging/mq:latest
icr.io/ibm-messaging/mq:latest
root@middleware2[19:43:02]:/root#docker images
REPOSITORY                TAG       IMAGE ID       CREATED       SIZE
icr.io/ibm-messaging/mq   latest    0a26aa49036c   4 weeks ago   1.67GB
root@middleware2[19:43:06]:/root#cle
```



```
root@middleware2[19:43:08]:/root#docker run -e LICENSE=accept -e MQ_QMGR_NAME=Tomcat9QM -p 1414:1414 -p 9443:9443 -d -e MQ_APP_PASSWORD=Tomca
t9 -e MQ_ADMIN_PASSWORD=Tomcat9 --name MQV9 icr.io/ibm-messaging/mq:latest
ff61c00de3951d3ebf156aaa53e67ab9fc3263fdd778951504171f3752ff60bb
root@middleware2[19:43:28]:/root#docker ps
CONTAINER ID   IMAGE                            COMMAND            CREATED          STATUS         PORTS
                                                          NAMES
ff61c00de395   icr.io/ibm-messaging/mq:latest   "runmqdevserver"   14 seconds ago   Up 7 seconds   0.0.0.0:1414->1414/tcp, :::1414->1414/tcp,
 0.0.0.0:9443->9443/tcp, :::9443->9443/tcp, 9157/tcp   MQV9
root@middleware2[19:43:35]:/root#
```

2) Log into container and check MQ Details

```
doker exec ff61c00de395 -it /bin/bash
```

dspmpver -: Display MQ version

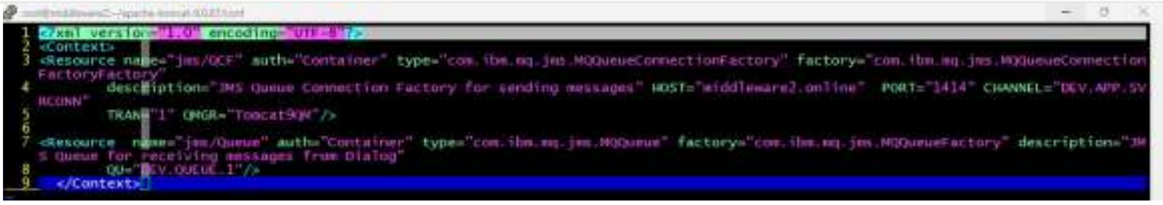
dspmq -: Display avaliable QueueManager and status

runmqc Commands are used to check MQ objects running inside QueueManager

3) Download Tomcat and check the start-up

4) Create a file context.xml in /root/apache-tomcat/conf and add below stanza for resource configuration. Queue connection Factory and Queue

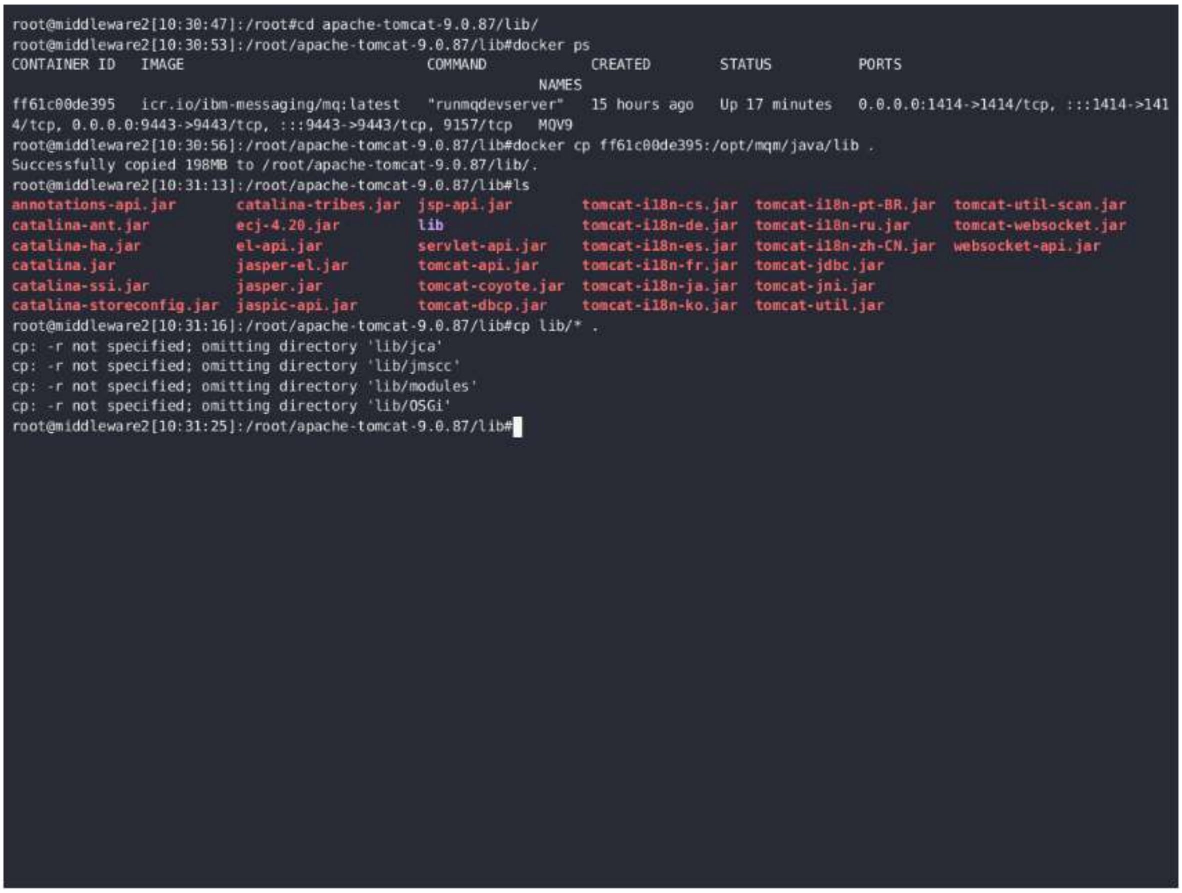
```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource name="jms/QCF" auth="Container"
type="com.ibm.mq.jms.MQQueueConnectionFactory"
factory="com.ibm.mq.jms.MQQueueConnectionFactoryFactory"
        description="JMS Queue Connection Factory for sending messages"
HOST="middleware2.online"  PORT="1414" CHANNEL="DEV.APP.SVRCONN"
        TRAN="1" QMGR="TomcatQM"/>

<Resource  name="jms/Queue" auth="Container" type="com.ibm.mq.jms.MQQueue"
factory="com.ibm.mq.jms.MQQueueFactory" description="JMS Queue for
receiving messages from Dialog"
        QU="DEV.QUEUE.1"/>
  </Context>
```



```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
<Resource name="jms/QCF" auth="Container" type="com.ibm.mq.jms.MQQueueConnectionFactory" factory="com.ibm.mq.jms.MQQueueConnection
FactoryFactory"
        description="JMS Queue Connection Factory for sending messages" HOST="middleware2.online"  PORT="1414" CHANNEL="DEV.APP.SV
RCONN"
        TRAN="1" QMGR="Tomcat9QM"/>

<Resource  name="jms/Queue" auth="Container" type="com.ibm.mq.jms.MQQueue" factory="com.ibm.mq.jms.MQQueueFactory" description="JM
S Queue for receiving messages from Dialog"
        QU="DEV.QUEUE.1"/>
  </Context>
```

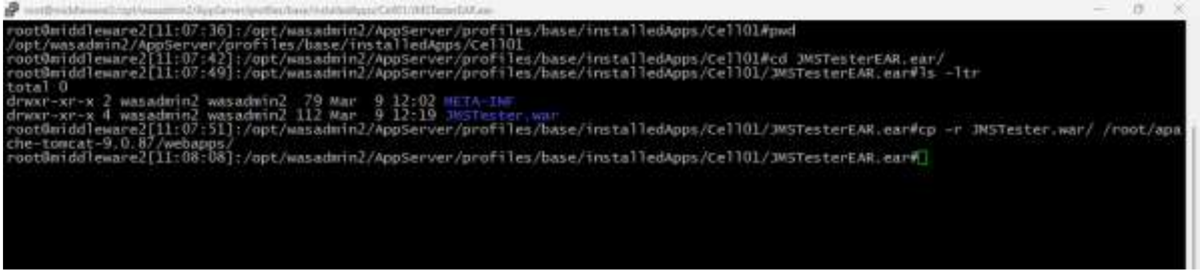
5) Below Jar files are required to connect to MQ successfully. Copy these jar files from running container to Tomcat Lib directory.

com.ibm.mq.allclient.jar
com.ibm.mq.commonservices.jar
com.ibm.mq.headers.jar
com.ibm.mq.jakarta.client.jar
com.ibm.mq.jar
com.ibm.mq.jmqi.jar
com.ibm.mqjms.jar
com.ibm.mq.pcf.jar
com.ibm.mq.tools.ras.jar
fscontext.jar
com.ibm.mq.traceControl.jar
jackson-annotations.jar
jackson-core.jar
jackson-databind.jar
jakarta.jms-api.jar
jms.jar
org.json.jar
providerutil.jar

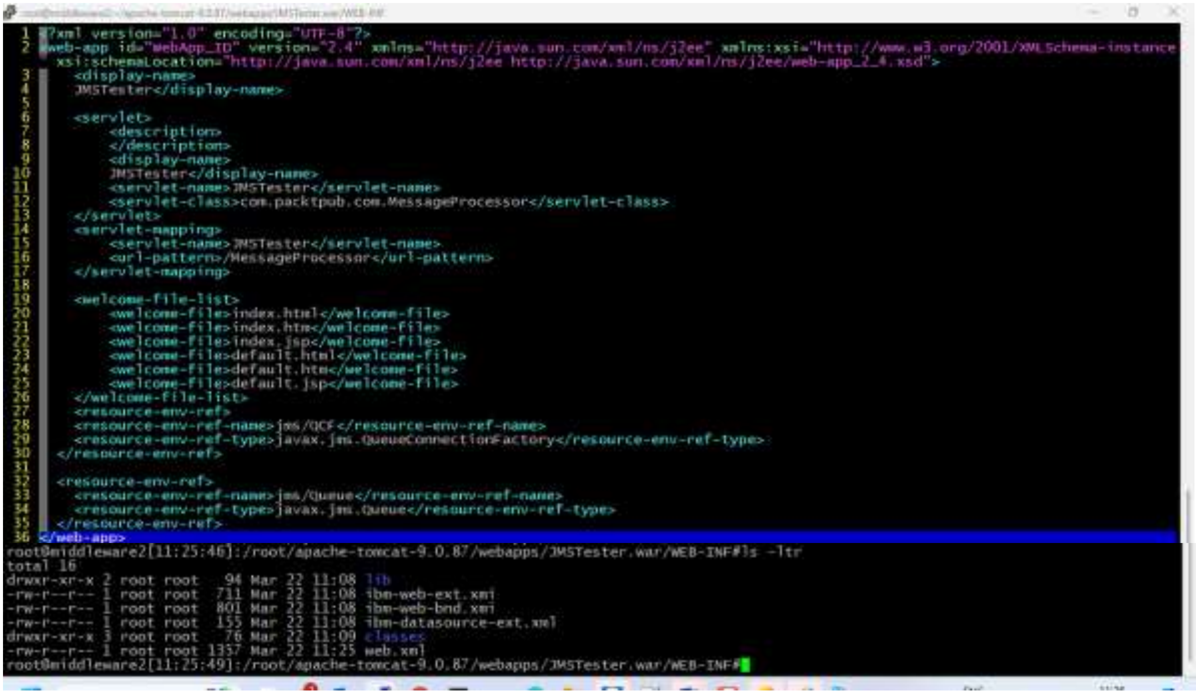


```
root@middleware2[10:30:47]:/root#cd apache-tomcat-9.0.87/lib/
root@middleware2[10:30:53]:/root/apache-tomcat-9.0.87/lib#docker ps
CONTAINER ID   IMAGE                              COMMAND              CREATED        STATUS          PORTS
                                                                     NAMES
ff61c00de395   icr.io/ibm-messaging/mq:latest   "runmqdevserver"   15 hours ago   Up 17 minutes   0.0.0.0:1414->1414/tcp, :::1414->141
4/tcp, 0.0.0.0:9443->9443/tcp, :::9443->9443/tcp, 9157/tcp   MQV9
root@middleware2[10:30:56]:/root/apache-tomcat-9.0.87/lib#docker cp ff61c00de395:/opt/mqm/java/lib .
Successfully copied 198MB to /root/apache-tomcat-9.0.87/lib/.
root@middleware2[10:31:13]:/root/apache-tomcat-9.0.87/lib#ls
annotations-api.jar      catalina-tribes.jar  jsp-api.jar         tomcat-i18n-cs.jar  tomcat-i18n-pt-BR.jar  tomcat-util-scan.jar
catalina-ant.jar         ecj-4.20.jar         lib                 tomcat-i18n-de.jar  tomcat-i18n-ru.jar     tomcat-websocket.jar
catalina-ha.jar          el-api.jar           servlet-api.jar     tomcat-i18n-es.jar  tomcat-i18n-zh-CN.jar  websocket-api.jar
catalina.jar             jasper-el.jar        tomcat-api.jar      tomcat-i18n-fr.jar  tomcat-jdbc.jar
catalina-ssi.jar         jasper.jar           tomcat-coyote.jar   tomcat-i18n-ja.jar  tomcat-jni.jar
catalina-storeconfig.jar jaspic-api.jar       tomcat-dbcp.jar     tomcat-i18n-ko.jar  tomcat-util.jar
root@middleware2[10:31:16]:/root/apache-tomcat-9.0.87/lib#cp lib/* .
cp: -r not specified; omitting directory 'lib/jca'
cp: -r not specified; omitting directory 'lib/jmscc'
cp: -r not specified; omitting directory 'lib/modules'
cp: -r not specified; omitting directory 'lib/OSGi'
root@middleware2[10:31:25]:/root/apache-tomcat-9.0.87/lib#
```

6) Copy Test Application deployed in WebSphere Application server 9 to /root/apache-tomcat/webapps and update web.xml

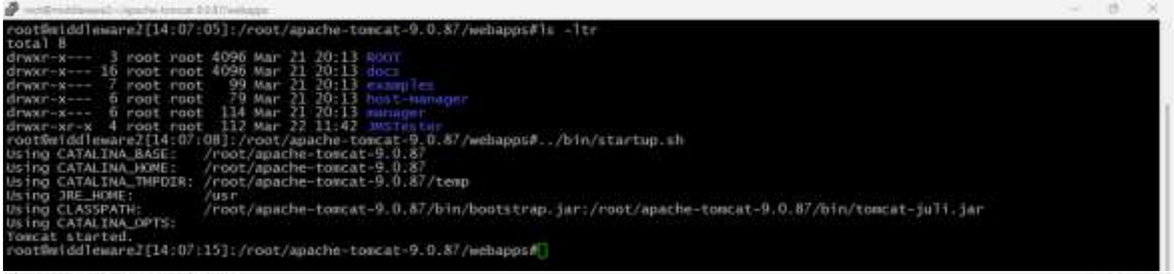


```
root@middleware2[11:07:36]:/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01#pwd
/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01
root@middleware2[11:07:42]:/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01#cd JMSTesterEAR.ear/
root@middleware2[11:07:49]:/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01/JMSTesterEAR.ear#ls -ltr
total 0
drwxr-xr-x 2 wasadmin2 wasadmin2  79 Mar  9 12:02 META-INF
drwxr-xr-x 4 wasadmin2 wasadmin2 112 Mar  9 12:19 JMSTester.war
root@middleware2[11:07:51]:/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01/JMSTesterEAR.ear#cp -r JMSTester.war/ /root/apa
che-tomcat-9.0.87/webapps/
root@middleware2[11:08:08]:/opt/wasadmin2/AppServer/profiles/base/installedApps/Cell01/JMSTesterEAR.ear#
```

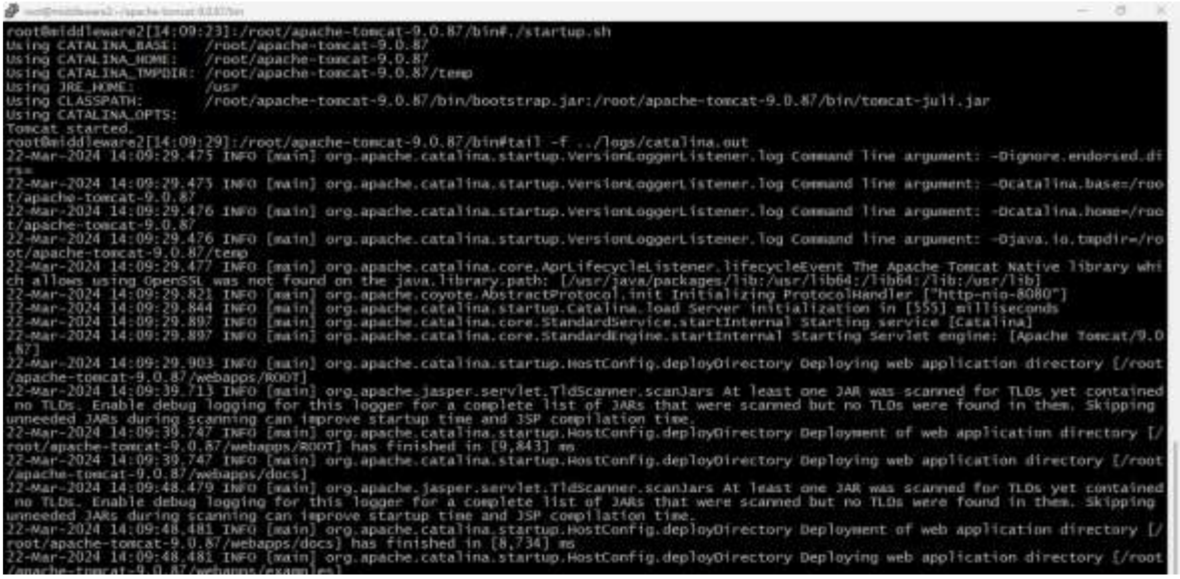


```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="WebApp_ID" version="2.4" xmlns="http://java.sun.com/xml/ns/j2ee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
    <display-name>
    JMSTester</display-name>

    <servlet>
        <description>
        </description>
        <display-name>
        JMSTester</display-name>
        <servlet-name>JMSTester</servlet-name>
        <servlet-class>com.packtpub.com.MessageProcessor</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>JMSTester</servlet-name>
        <url-pattern>/MessageProcessor</url-pattern>
    </servlet-mapping>

    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</welcome-file>
    </welcome-file-list>
    <resource-env-ref>
    <resource-env-ref-name>jms/QCF</resource-env-ref-name>
    <resource-env-ref-type>javax.jms.QueueConnectionFactory</resource-env-ref-type>
  </resource-env-ref>

  <resource-env-ref>
    <resource-env-ref-name>jms/Queue</resource-env-ref-name>
    <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
  </resource-env-ref>
</web-app>
root@middleware2[11:25:46]:/root/apache-tomcat-9.0.87/webapps/JMSTester.war/WEB-INF#ls -ltr
total 16
drwxr-xr-x 2 root root   94 Mar 22 11:08 lib
-rw-r--r-- 1 root root  711 Mar 22 11:08 ibm-web-ext.xmi
-rw-r--r-- 1 root root  801 Mar 22 11:08 ibm-web-bnd.xmi
-rw-r--r-- 1 root root  155 Mar 22 11:08 ibm-datasource-ext.xml
drwxr-xr-x 3 root root   76 Mar 22 11:09 classes
-rw-r--r-- 1 root root 1357 Mar 22 11:25 web.xml
root@middleware2[11:25:49]:/root/apache-tomcat-9.0.87/webapps/JMSTester.war/WEB-INF#
```

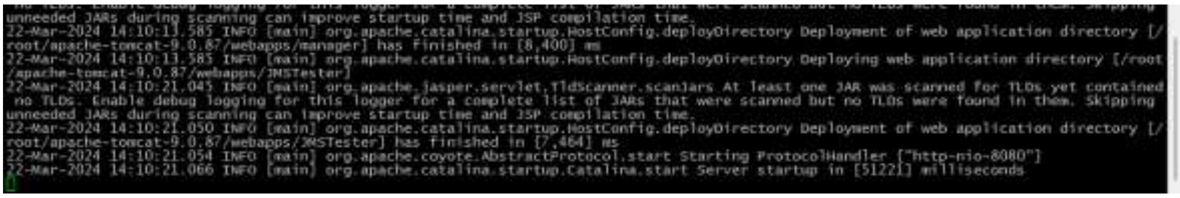
7)Rename the application name to JMSTester , Start Tomcat and check log file and access application



```
root@middleware2[14:07:05]:/root/apache-tomcat-9.0.87/webapps#ls -ltr
total 8
drwxr-x---  3 root root 4096 Mar 21 20:13 ROOT
drwxr-x--- 16 root root 4096 Mar 21 20:13 docs
drwxr-x---  7 root root   99 Mar 21 20:13 examples
drwxr-x---  6 root root   79 Mar 21 20:13 host-manager
drwxr-x---  6 root root  114 Mar 21 20:13 manager
drwxr-xr-x  4 root root  112 Mar 22 11:42 JMSTester
root@middleware2[14:07:08]:/root/apache-tomcat-9.0.87/webapps#../bin/startup.sh
Using CATALINA_BASE:   /root/apache-tomcat-9.0.87
Using CATALINA_HOME:   /root/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /root/apache-tomcat-9.0.87/bin/bootstrap.jar:/root/apache-tomcat-9.0.87/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@middleware2[14:07:15]:/root/apache-tomcat-9.0.87/webapps#
```



```
root@middleware2[14:09:23]:/root/apache-tomcat-9.0.87/bin#./startup.sh
Using CATALINA_BASE:   /root/apache-tomcat-9.0.87
Using CATALINA_HOME:   /root/apache-tomcat-9.0.87
Using CATALINA_TMPDIR: /root/apache-tomcat-9.0.87/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /root/apache-tomcat-9.0.87/bin/bootstrap.jar:/root/apache-tomcat-9.0.87/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
root@middleware2[14:09:29]:/root/apache-tomcat-9.0.87/bin#tail -f ../logs/catalina.out
22-Mar-2024 14:09:29.475 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dignore.endorsed.di
rs=
22-Mar-2024 14:09:29.475 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.base=/roo
t/apache-tomcat-9.0.87
22-Mar-2024 14:09:29.476 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Dcatalina.home=/roo
t/apache-tomcat-9.0.87
22-Mar-2024 14:09:29.476 INFO [main] org.apache.catalina.startup.VersionLoggerListener.log Command line argument: -Djava.io.tmpdir=/ro
ot/apache-tomcat-9.0.87/temp
22-Mar-2024 14:09:29.477 INFO [main] org.apache.catalina.core.AprLifecycleListener.lifecycleEvent The Apache Tomcat Native library whi
ch allows using OpenSSL was not found on the java.library.path: [/usr/java/packages/lib:/usr/lib64:/lib64:/lib:/usr/lib]
22-Mar-2024 14:09:29.821 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["http-nio-8080"]
22-Mar-2024 14:09:29.844 INFO [main] org.apache.catalina.startup.Catalina.load Server initialization in [555] milliseconds
22-Mar-2024 14:09:29.897 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
22-Mar-2024 14:09:29.897 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet engine: [Apache Tomcat/9.0
.87]
22-Mar-2024 14:09:29.903 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [/root
/apache-tomcat-9.0.87/webapps/ROOT]
22-Mar-2024 14:09:39.713 INFO [main] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained
 no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping
unneeded JARs during scanning can improve startup time and JSP compilation time.
22-Mar-2024 14:09:39.747 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [/
root/apache-tomcat-9.0.87/webapps/ROOT] has finished in [9,843] ms
22-Mar-2024 14:09:39.747 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [/root
/apache-tomcat-9.0.87/webapps/docs]
22-Mar-2024 14:09:48.479 INFO [main] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained
 no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping
unneeded JARs during scanning can improve startup time and JSP compilation time.
22-Mar-2024 14:09:48.481 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [/
root/apache-tomcat-9.0.87/webapps/docs] has finished in [8,734] ms
22-Mar-2024 14:09:48.481 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [/root
/apache-tomcat-9.0.87/webapps/examples]
```



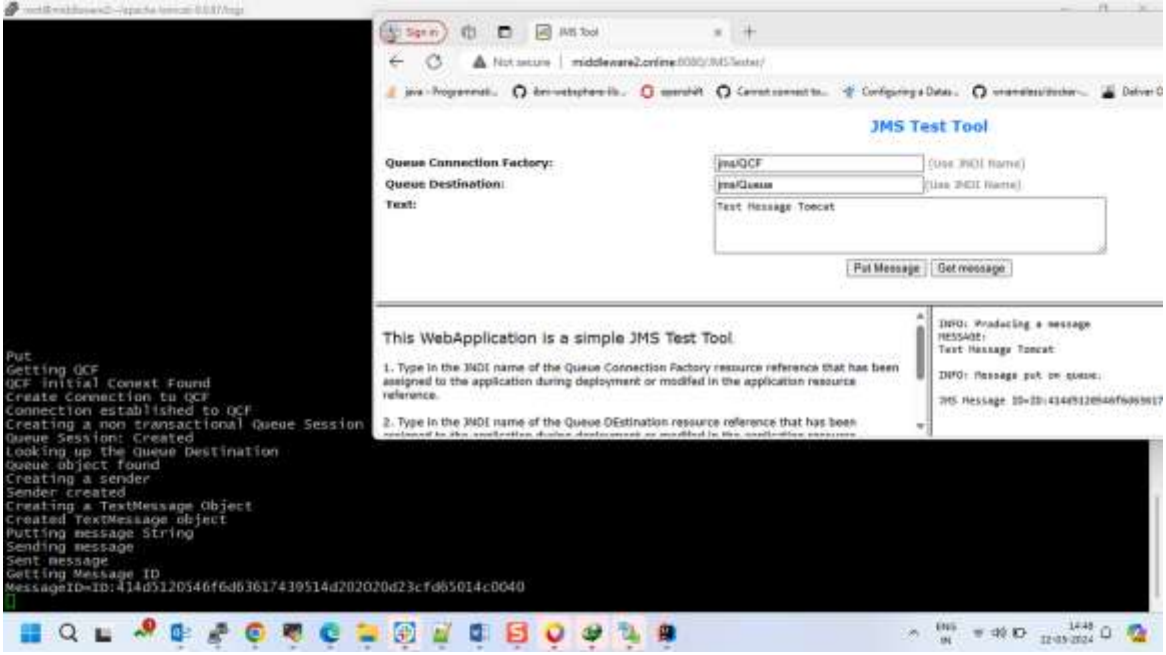
```
unneeded JARs during scanning can improve startup time and JSP compilation time.
22-Mar-2024 14:10:13.585 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [/
root/apache-tomcat-9.0.87/webapps/manager] has finished in [8,400] ms
22-Mar-2024 14:10:13.585 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [/root
/apache-tomcat-9.0.87/webapps/JMSTester]
22-Mar-2024 14:10:21.045 INFO [main] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained
 no TLDs. Enable debug logging for this logger for a complete list of JARs that were scanned but no TLDs were found in them. Skipping
unneeded JARs during scanning can improve startup time and JSP compilation time.
22-Mar-2024 14:10:21.050 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [/
root/apache-tomcat-9.0.87/webapps/JMSTester] has finished in [7,464] ms
22-Mar-2024 14:10:21.054 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
22-Mar-2024 14:10:21.066 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [51221] milliseconds
```

Connect to MQ container and execute the below mqsc commands to disable Password check while connecting to MQ

ALTER AUTHINFO(SYSTEM.DEFAULT.AUTHINFO.IDPWOS) AUTHTYPE(IDPWOS) CHCKCLNT(NONE) CHCKLOCL(NONE)

REFRESH SECURITY

8) Access application, put and get messages



JMS Test Tool

Queue Connection Factory: jms/QCF (Use JNDI Name)

Queue Destination: jms/Queue (Use JNDI Name)

Text: Test Message Tomcat

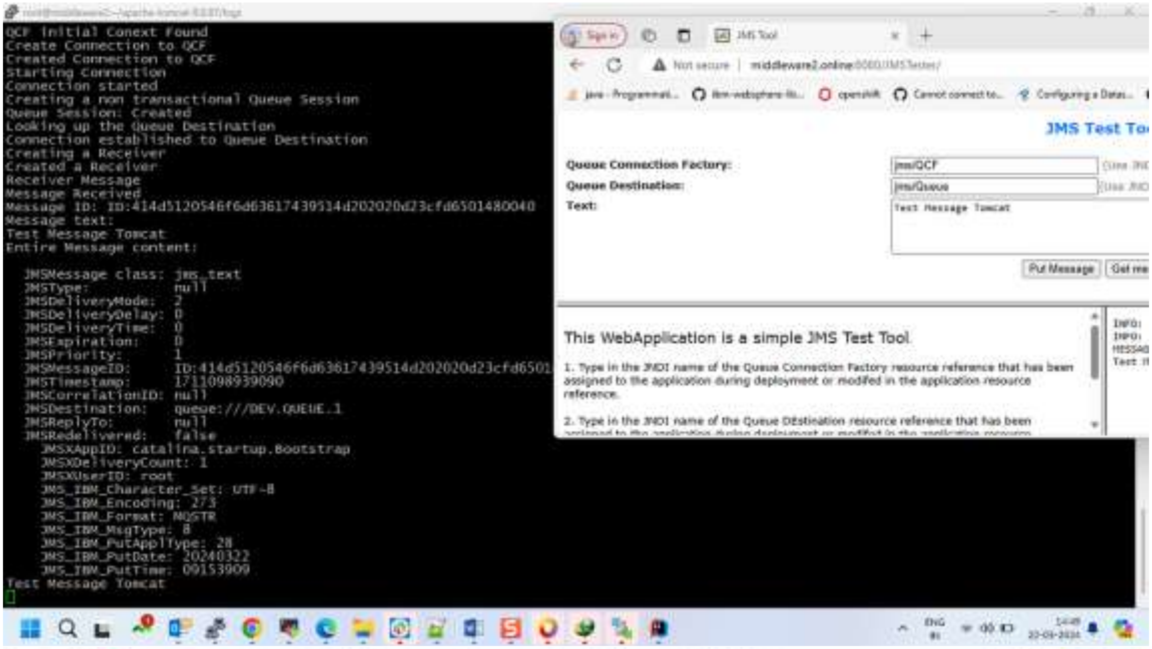
This WebApplication is a simple JMS Test Tool

1. Type in the JNDI name of the Queue Connection Factory resource reference that has been assigned to the application during deployment or modifed in the application resource reference.

2. Type in the JNDI name of the Queue DEstination resource reference that has been

```
INFO: Producing a message
MESSAGE:
Test Message Tomcat

INFO: Message put on queue:

JMS Message ID=ID:414d5120546f6d63617
```

```
Put
Getting QCF
QCF initial Conext Found
Create Connection to QCF
Connection established to QCF
Creating a non transactional Queue Session
Queue Session: Created
Looking up the Queue Destination
Queue object found
Creating a sender
Sender created
Creating a TextMessage Object
Created TextMessage object
Putting message String
Sending message
Sent message
Getting Message ID
MessageID=ID:414d5120546f6d63617439514d202020d23cfd65014c0040
```



```
QCF initial Conext Found
Create Connection to QCF
Created Connection to QCF
Starting Connection
Connection started
Creating a non transactional Queue Session
Queue Session: Created
Looking up the Queue Destination
Connection established to Queue Destination
Creating a Receiver
Created a Receiver
Receiver Message
Message Received
Message ID: ID:414d5120546f6d636174395l4d202020d23cfd6501480040
Message text:
Test Message Tomcat
Entire Message content:

  JMSMessage class: jms_text
  JMSType:          null
  JMSDeliveryMode:  2
  JMSDeliveryDelay: 0
  JMSDeliveryTime:  0
  JMSExpiration:    0
  JMSPriority:      1
  JMSMessageID:     ID:414d5120546f6d636174395l4d202020d23cfd6501
  JMSTimestamp:     1711098939090
  JMSCorrelationID: null
  JMSDestination:   queue:///DEV.QUEUE.1
  JMSReplyTo:       null
  JMSRedelivered:   false
    JMSXAppID: catalina.startup.Bootstrap
    JMSXDeliveryCount: 1
    JMSXUserID: root
    JMS_IBM_Character_Set: UTF-8
    JMS_IBM_Encoding: 273
    JMS_IBM_Format: MQSTR
    JMS_IBM_MsgType: 8
    JMS_IBM_PutApplType: 28
    JMS_IBM_PutDate: 20240322
    JMS_IBM_PutTime: 09153909
Test Message Tomcat
```

Queue Connection Factory: jms/QCF

Queue Destination: jms/Queue

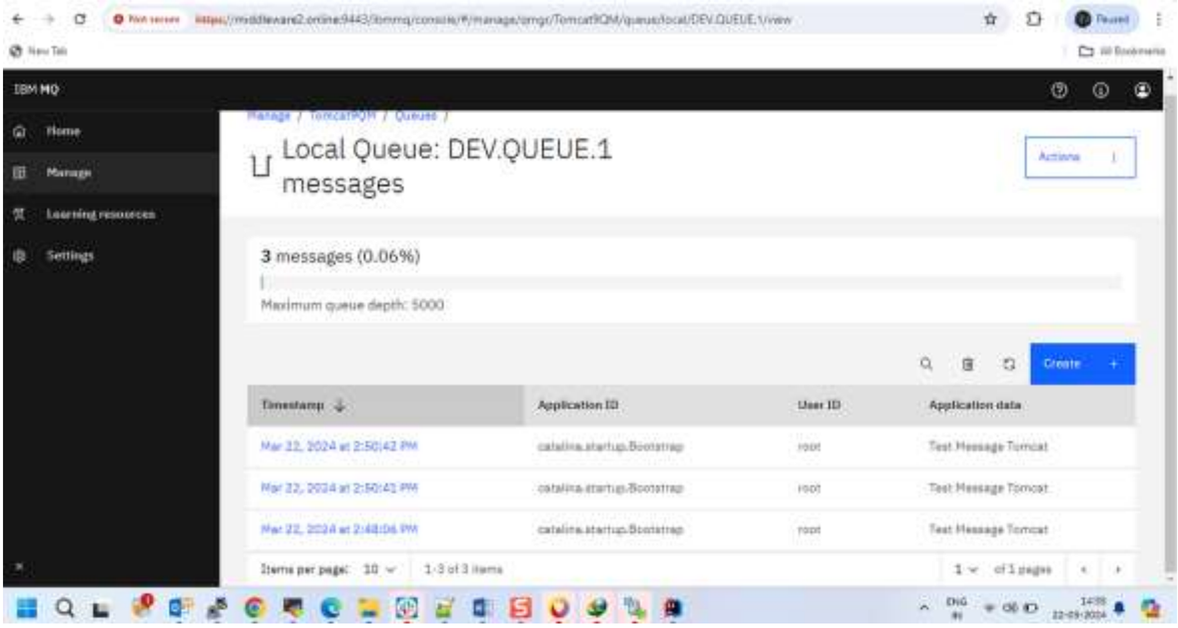
Text: Test Message Tomcat

This WebApplication is a simple JMS Test Tool

1. Type in the JNDI name of the Queue Connection Factory resource reference that has been assigned to the application during deployment or modifed in the application resource reference.

2. Type in the JNDI name of the Queue DEstination resource reference that has been

```
INFO:
INFO:
MESSAG
Text M
```



https://middleware2.online:9443/ibmmq/console/#/manage/qmgr/Tomcat9QM/queue/local/DEV.QUEUE.1/view
New Tab
IBM MQ
Home
Manage
Learning resources
Settings
Manage / Tomcat9QM / Queues /
Actions
Create +

Local Queue: DEV.QUEUE.1 messages

3 messages (0.06%)

Maximum queue depth: 5000

Timestamp ↓	Application ID	User ID	Application data
Mar 22, 2024 at 2:50:42 PM	catalina.startup.Bootstrap	root	Test Message Tomcat
Mar 22, 2024 at 2:50:41 PM	catalina.startup.Bootstrap	root	Test Message Tomcat
Mar 22, 2024 at 2:48:06 PM	catalina.startup.Bootstrap	root	Test Message Tomcat

Items per page: 10 1-3 of 3 items 1 of 1 pages