|  |
| --- |
| Deployment Guide |
| MOIC Production Build Release |

Contents

[Prerequisite 1](#_Toc60414991)

[Deployment Steps: 2](#_Toc60414992)

# Prerequisite

* Software List:
* Angular - 9.1.12
* Node - latest version
* Open Java 1.8
* Tomcat 9.0.40
* Server Access:

One should have access to ssh to production server and then to switch sudo user

Host name: *usawiplcweapp03*

Make sure you have permission to impersonate (sudo su – moicadm) user: moicadm

* Use PAM client to ssh to above mentioned server
* Switch to moicadm sudo user(sudo su – moicadm)
* List of ports firewall request needs to be opened for following on the server:

1. 8443 : to access application through HTTPS
2. 5432: Application needs this PostgreSQL database port to communicate.

First time set up/installation for environment readiness

Once you are able to access sudo user, please verify below

* Java 8 is installed
* Tomcat is placed at location /apps/moic/tomcat/apache-tomcat-\*
* Directory hierarchy creation

Now you need to create directory structure which is required for the application to process inbound and outbound jobs.

Please create directory hierarchy as mentioned below.

1. **create directories as below:**

Navigate to /apps/moic with moicadm user and then create /moic\_integration directory as parent to all integration directories then create /joor\_orders, /customer\_master and /product\_master directories within /moic\_integration directory.

Navigate to each integration directories and create /in, /out, /archive, /failed directories as shown below.

/apps/moic

/moic\_integration

/joor\_orders

/in

/out

/archive

/failed

/customer\_master

/in

/out

/archive

/failed

/product\_master

/in

/out

/archive

/failed

1. **Change permission of the above created directories**

Navigate to /apps/moic with moicadm user and change permission of moic\_integration and it’s child directories by executing below command.

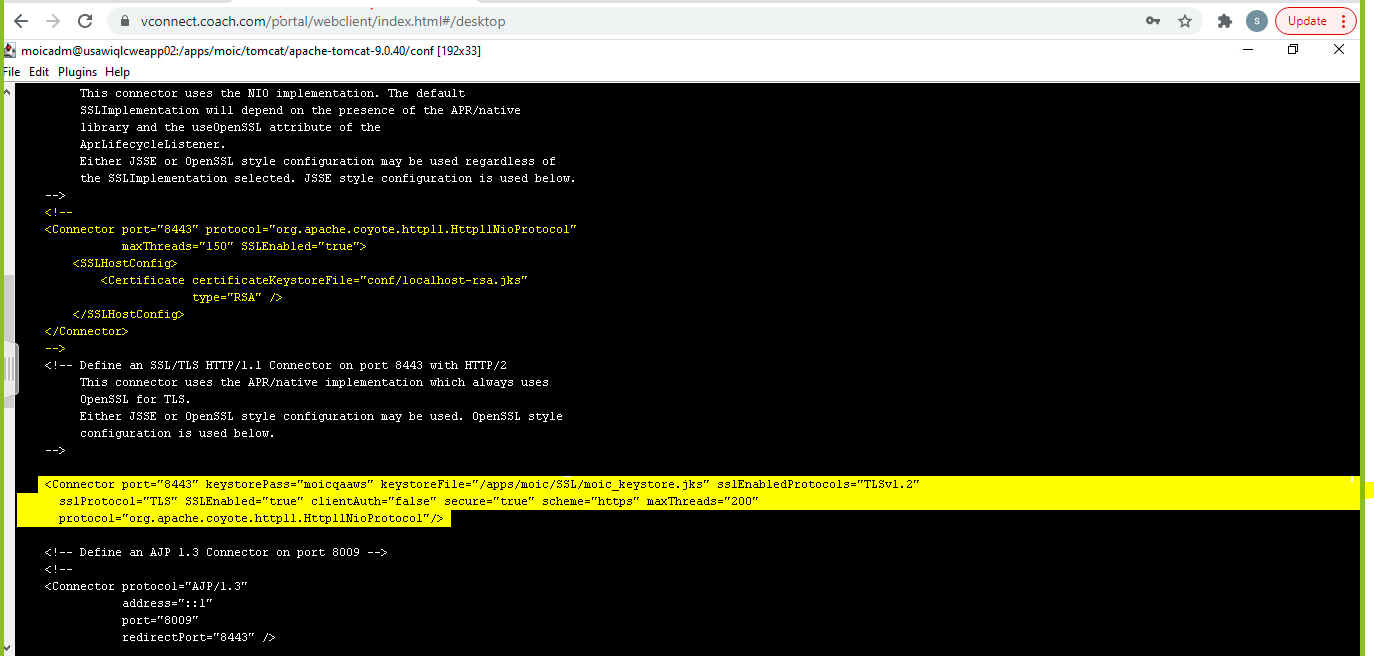
**chmod 777 /moic\_integration/\***

1. **Place scheduler file provided by MOIC team**

Moic team should have provide you **moic\_prd\_scheduler.sh** file, Please copy it to this location **/apps/moic/moic\_integration/** and change it’s permission with **chmod 777 moic\_prd\_scheduler.sh** command.

* Configure SSL / 8443 port to enable application on https

1. Please login to production environment with PAM client and then switch to sudo user (sudo su – moicadm) and navigate to tomcat conf directory /apps/moic/tomcat/apache-tomcat-\*/conf .
2. Open server.xml using vi editor vi server.xml and look for a connector tag for 8443 port, Please refer below screenshot.



1. Please insert below snippet after commented 8443 Connector tag

|  |
| --- |
| <Connector port="8443" keystorePass="<keystore\_password>" keystoreFile="/apps/moic/SSL/moic\_keystore.jks" sslEnabledProtocols="TLSv1.2" sslProtocol="TLS" SSLEnabled="true" clientAuth="false" secure="true" scheme="https" maxThreads="200" protocol="org.apache.coyote.http11.Http11NioProtocol"/> |

1. Please make sure moic\_keystore.jks exist on the specified path.
2. If you have not been provided with keystorePass, Please reach out to Linux team who has generated this .jks at the time of environment setup.

* Configure External property file for Production Environment

1. Navigate to tomcat bin directory /apps/moic/tomcat/apache-tomcat-\*/bin and create setenv.sh (Linux server) or setenv.bat (windows server).
2. Put following line in setenv file
3. export CATALINA\_OPTS="-Dspring.config.location=**<path to external property file>**/external\_property\_file\_name"
4. Ex: export CATALINA\_OPTS="-Dspring.config.location=/apps/moic/external\_props/application-dev.yml"
5. Navigate to **<path to external property file>** and copy respective environment property file received from MOIC team.
6. Edit this property file and provide username and password under all datasource sections.

**NOTE:** This username and passwords are db username and passwords which is required by application to interact with the database.

1. Please restart the tomcat server in order to verify that application is communicating with database before we provide the encrypted password within property file.
2. On successful server start up please shut down the server and follow below instructions to encrypt the plain password kept in the property file.

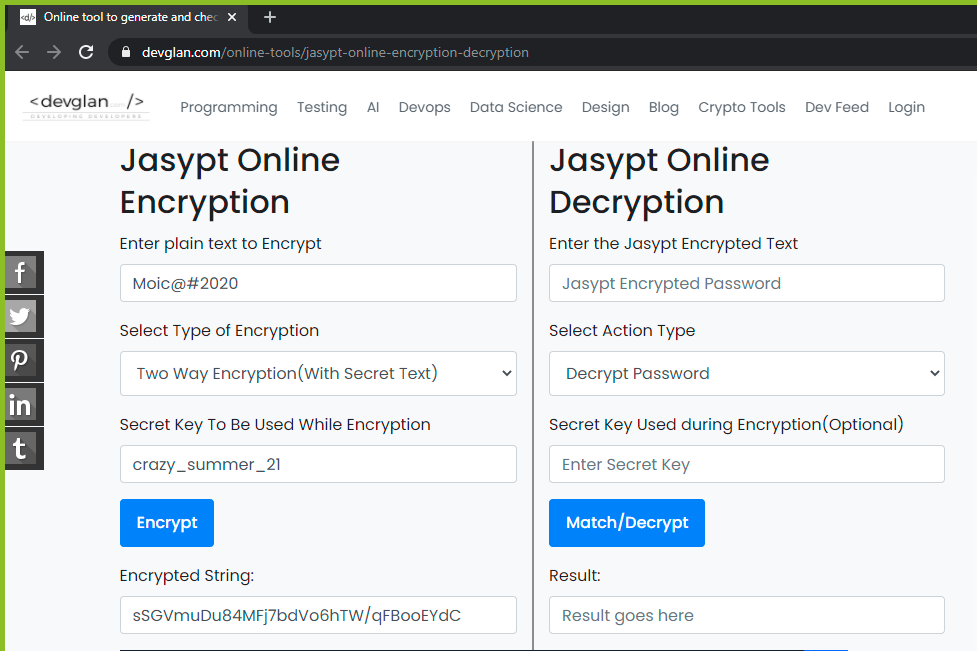
* Jasypt database password encryption for external property file

You can encrypt password using Jasypt using the given link <https://www.devglan.com/online-tools/jasypt-online-encryption-decryption> .

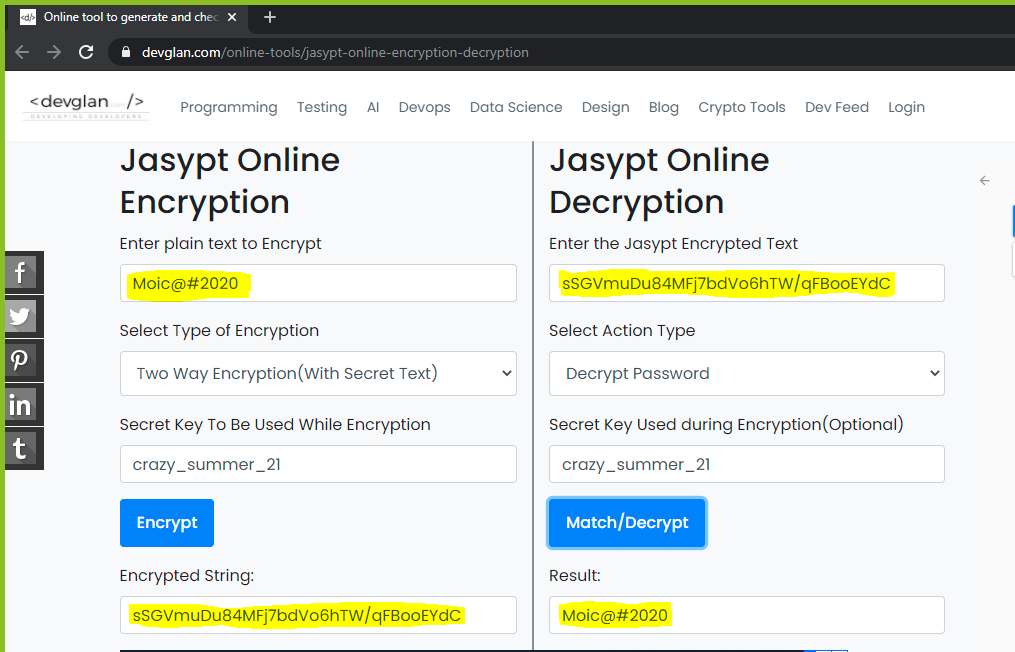
You will need following thing.

* <Password> (will be provided by Database team)
* <Secret Key> (will be provided by MOIC team)

1. Click on the given link above.
2. Please enter plain password in *Enter plain text to Encrypt* input field, Select *Select Type of Encryption =* Two Way Encryption (With Secret Text) and enter Secret Key in *Secret Key To Be Used While Encryption*  input field as displayed in below screenshot.



1. This will generate new Encrypted String as shown in image above, Please decrypt the same in order to verify it got encrypted properly. Please refer below screenshot.



4.Get the generated Encrypted String and paste it into application-<env>.yml file in place of password within ENC(Encrypted String).

For ex: spring:

datasource-internal:

password: ENC(Encrypted String)

* Production Database Prerequisites

DB team will provide the connection details, which needs to be inserted into the external property file.

There will be 3 schema moic, moic\_inbound, moic\_outbound respectively.

DB team will provide different user and their credentials for each schema with all permissions.

Using these db users/credentials MOIC application will interact with database.

There is a requirement to preload-ed/insert data into look up tables in order to make the application functional.The following tables needs to be pre populated:

moic\_user

roles

currencyconversion

segment

Channel

* Azzure AD

As per application's technical design user needs to be added to Azzure AD group of specific environment.

There is a different team handling Azzure AD, in case of addition or removal of any user please reach out to Kevin

The same user needs to be exist in the MOIC application database moic\_user table along with role association in roles table.

* Details of Tidal server jobs.

All details of Tidal jobs will be provided in separate file by MOIC team. We can run this job manually as mentioned in the file and we can run those jobs from Tidal server as well. The name, group and other details of jobs are mentioned in Tidal Master Jobs file.

# Spring app Deployment Steps:

1. **Connect via PAM client to MOIC Production Server (*usawiplcweapp03*)**
2. Transfer the War file provided by MOIC team using the same user you are logged in with , once you transfer the war file please change the permissions for that using command chmod 777 Moic-api-0.0.1.war .
3. Move this war file to the shared location /tmp which can be accessed by moicadm sudo user.
4. *Switch the user to sudo su – moicadm*

**Note: All following steps should be executed via moicadm user only**

1. **Stop Tomcat using moicadm user**
2. To stop tomcat, execute this command sudo systemctl stop tomcat
3. Please navigate to /apps/moic/tomcat/apache-tomcat-\*/logs and verify logs to make sure the server is stopped tail -f catalina.out
4. **Copy Moic-api-0.0.1.war Files to webapps of Tomcat Directory using moicadm user**
5. Navigate to /apps/moic/tomcat/apache-tomcat-\*/webapps/
6. Move existing Moic-api-0.0.1.war file to /tmp folder with different name Moic-api-0.0.1.war.DD-MM-YYYY in order to create a rollback version.
7. Copy Moic-api-0.0.1.war file from /tmp to /apps/moic/tomcat/apache-tomcat-\*/webapps/
8. **Start Tomcat using moicadm user**
9. To start tomcat, execute this command sudo systemctl start tomcat
10. Please navigate to /apps/moic/tomcat/apache-tomcat-\*/logs and verify logs using command tail -f catalina.out to make sure the server is started and war deployed successfully.

# Angular app Deployment Steps:

1. **Connect via PAM client to MOIC Production Server (*usawiplcweapp03*)**
   1. Transfer the Zip file provided by MOIC team using the same user you are logged in with , once you transfer the zip file please change the permissions for that using command chmod 777 Moic-<env>.zip .
   2. Move this zip file to the shared location /tmp which can be accessed by moicadm sudo user.
   3. Switch the user to sudo su – moicadm
   4. Unzip the zip file using unzip Moic-<env>.zip command.
   5. Unzipping this will create moic folder so please remove/rename any existing folder with same name in order to avoid any confusions.

**Note: All following steps should be executed via moicadm user only**

1. **Copy Files to webapps of Tomcat Directory using moicadm user**
2. Navigate to /apps/moic/tomcat/apache-tomcat-\*/webapps/
3. Move existing moic folder to /tmp folder with different name as moic-DD-MM-YYYY in order to create a rollback version.
4. Move moic folder from /tmp to /apps/moic/tomcat/apache-tomcat-\*/webapps/ Using command mv /tmp/moic /apps/moic/tomcat/apache-tomcat-\*/webapps/
5. Please navigate to /apps/moic/tomcat/apache-tomcat-\*/logs and verify logs using command tail -f catalina.out to make sure the angular is deployed successfully.

Capture logs and provide or attach to ticket so we can verify deployment or any production issues

1. Please download tomcat logs from */apps/moic/tomcat/apache-tomcat-\*/logs/catalina.out*
2. Please download Moic api Jobs logs from */apps/moic/tomcat/apache-tomcat-\*/moic-logs/*MoicApi\_Jobs.yyyy-MM-dd.n.log
3. Upload Catalina.out and *moic-logs/*MoicApi\_Jobs.yyyy-MM-dd.n.log files to ticket or send us mail moicteam@cignex.com