# Steps to install camunda

1. Download <https://camunda.org/release/camunda-bpm/wildfly10/7.8/camunda-bpm-wildfly10-7.8.0.zip> (please get the correct version of camunda)
2. Extract zip file in /opt/camunda folder in azure VM.
3. Navigate to /opt/camunda/server/wildfly-10.1.0.Final/standalone/configuration/standalone.xml file.
4. Change 127.0.0.1 ip address with internal ip of azure VM (search for 127.0.0.1 in standalone.xml)

## Steps to Connect to MSSQL:

1. To connect to mssql, create directory or navigate to /opt/camunda/server/wildfly-10.1.0.Final/modules/system/layers/base/com/microsoft/sqlserver/main
2. Download and add mssql-jdbc-6.4.0.jre8.jar file in above directory
3. Create module.xml file with below content.

**<?xml version="1.0" encoding="UTF-8"?>**

**<module xmlns="urn:jboss:module:1.3" name="com.microsoft.sqlserver">**

**<resources>**

**<resource-root path="mssql-jdbc-6.4.0.jre8.jar"/>**

**</resources>**

**<dependencies>**

**<module name="javax.api"/>**

**<module name="javax.transaction.api"/>**

**<module name="javax.xml.bind.api" />**

**</dependencies>**

**</module>**

1. In Standalone.xml file under datasource section add below content

**<datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">**

**<connection-url>jdbc:sqlserver://<IP>:<Port>;databaseName=<DBNAME></connection-url>**

**<driver>mssql</driver>**

**<security>**

**<user-name><USERNAME></user-name>**

**<password><<PWD>></password>**

**</security>**

**</datasource>**

**<datasource jta="true" jndi-name="java:jboss/datasources/ProcessEngine" pool-name="ProcessEngine" enabled="true" use-java-context="true" use-ccm="true">**

**<connection-url>jdbc:sqlserver://<IP>:<Port>;databaseName=<DBNAME></connection-url>**

**<driver>mssql</driver>**

**<security>**

**<user-name><USERNAME></user-name>**

**<password><PWD></password>**

**</security>**

**<validation>**

**<valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.mssql.MSSQLValidConnectionChecker"/>**

**</validation>**

**</datasource>**

**<drivers>**

**<driver name="h2" module="com.h2database.h2">**

**<xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>**

**</driver>**

**<driver name="mssql" module="com.microsoft.sqlserver">**

**<driver-class>com.microsoft.sqlserver.jdbc.SQLServerDriver</driver-class>**

**</driver>**

**</drivers>**

## Enable SSL for Camunda on WildFly:

1. Place the .jks file on the camunda VM under configuration folder (/opt/camunda/server/wildfly-10.1.0.Final/standalone/configuration/sslcert)
2. open standalone.xml and under management option create custom security-realm with any name you want I have given SslRelam as shown below:

**<security-realm name="SslRealm">**

**<server-identities>**

**<ssl>**

**<keystore path="/opt/camunda/server/wildfly-10.1.0.Final/standalone/configuration/sslcert/\*\*\*\*\*\*\*.jks" keystore-password="\*\*\*\*\*\*\*\*\*" alias="\*\*\*\*\*\*\*"/>**

**</ssl>**

**</server-identities>**

**</security-realm>**

1. Under profile option expand undertow subsystem there are two main parts which are server and Servlet container configuration
2. Add https-listener under server for SslRealm created in step10

**<https-listener name="default-ssl" socket-binding="https" security-realm="SslRealm"/>**

1. Change Default WildfFly https port 8443 to 443. Under <socket-binding-group> change 8443 to 443

**<socket-binding name="https" port="443"/>**

## Camunda LDAP integration steps:

1. Add LDAP plugin in /opt/camunda/server/wildfly-10.1.0.Final/standalone/configuration/standalone.xml file as below under plugins section. Change the below properties with the valid LDAP details.

**<plugin>**

**<class>org.camunda.bpm.identity.impl.ldap.plugin.LdapIdentityProviderPlugin</class>**

**<properties>**

**<property name="serverUrl">ldaps://localhost:4334/</property>**

**<property name="acceptUntrustedCertificates">false</property>**

**<property name="managerDn">uid=jonny,ou=office-berlin,o=camunda,c=org</property>**

**<property name="managerPassword">s3cr3t</property>**

**<property name="baseDn">o=camunda,c=org</property>**

**<property name="userSearchBase">ou=employees</property>**

**<property name="userSearchFilter">(objectclass=person)</property>**

**<property name="userIdAttribute">uid</property>**

**<property name="userFirstnameAttribute">cn</property>**

**<property name="userLastnameAttribute">sn</property>**

**<property name="userEmailAttribute">mail</property>**

**<property name="userPasswordAttribute">userpassword</property>**

**<property name="groupSearchBase">ou=roles</property>**

**<property name="groupSearchFilter">(objectclass=groupOfNames)</property>**

**<property name="groupIdAttribute">cn</property>**

**<property name="groupNameAttribute">cn</property>**

**<property name="groupMemberAttribute">member</property>**

**<property name="sortControlSupported">false</property>**

**</properties>**

**</plugin>**

**<plugin>**

**<class>org.camunda.bpm.engine.impl.plugin.AdministratorAuthorizationPlugin</class>**

**<properties>**

**<property name="administratorUserName">admin</property>**

**</properties>**

**</plugin>**

## Steps to install AppD agent:

1. Download AppD java agent on to camunda VM
2. Navigate to /opt/camunda/server/wildfly-10.1.0.Final/bin and open standalone.conf file.
3. Add the following to the end of the standalone.conf file in the JAVA\_OPTS section.

JAVA\_OPTS="-Djava.util.logging.manager=org.jboss.logmanager.LogManager -Xbootclasspath/p:/opt/camunda/server/wildfly-10.1.0.Final/modules/system/layers/base/org/jboss/logmanager/main/jboss-logmanager-2.0.4.Final.jar"

1. Place the below argument in standalone.sh

export JAVA\_OPTS="$JAVA\_OPTS -javaagent:/opt/AppD/ver4.5.1.23676/javaagent.jar -Dappdynamics.agent.applicationName=EIAP\_NonProd -Dappdynamics.agent.tierName=camunda -Dappdynamics.agent.nodeName=eipcamundaqa1"

Ref link: <https://docs.appdynamics.com/display/PRO45/JBoss+and+Wildfly+Startup+Settings>

## Steps to enable basic authentication for camunda rest API’s:

1. By default camunda rest apis are not enabled with basic authentication to enable basic authentication extract camunda-engine-rest-7.8.0.war and uncomment below content from web.xml file as below

<!-- Http Basic Authentication Filter -->

<filter>

<filter-name>camunda-auth</filter-name>

<filter-class>

org.camunda.bpm.engine.rest.security.auth.ProcessEngineAuthenticationFilter

</filter-class>

<init-param>

<param-name>authentication-provider</param-name>

<param-value>org.camunda.bpm.engine.rest.security.auth.impl.HttpBasicAuthenticationProvider</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>camunda-auth</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

1. Now Camunda rest api’s are enabled with authentication if camunda is LDAP enabled credentials for rest api’s will be LDAP user credentials.

## Steps to create database and camunda tables:

1. Connect to sql server and get the camunda related tables script from /opt/camunda/sql/create folder where camunda is installed and execute below sql scripts

mssql\_engine\_7.8.0.sql

mssql\_identity\_7.8.0.sql

## Steps to Start Camunda:

1. Navigate to directory /opt/camunda and execute **./start-camunda.sh </dev/null >/dev/null 2>&1 &**

**NOTE: Please make sure to delete default workflows from camunda deployment folders. Like camunda-example-invoice-\*\*\*.war and camunda-h2-webapp-7.8.9-ee-\*\*\*.war.**