



Installazione di Bonita 7

Bundle pronto all'uso

Un bundle della versione corretta lo puoi trovare in:

```
smb://storage.bo.priv/RESOURCES/Bonita/7.1.4/
```

Installazione Completa

Utilizzeremo il bundle di Bonita 7 + tomcat 7 e database PostgreSQL 9.6 o inferiori. La guida ufficiale è consultabile [qui](#)

Dopo aver scaricato e scompattato il bundle in una cartella temporanea spostiamo l'intera struttura in /opt/

Successivamente creiamo un symlink alla cartella appena spostata chiamandolo bonita7

[Qui](#) possiamo trovare una guida sull'installazione di PostgreSQL per macchine CentOS/AlmaLinux

PostgreSQL

Se procedi con un'installazione nuova dopo aver installato ed inizializzato il database modifica il file /var/lib/pgsql/data/pg_hba.conf come segue per attivare l'autenticazione al database:

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		peer
# IPv4 local connections:					
host	all		all	127.0.0.1/32	md5
# IPv6 local connections:					
host	all		all	:::1/128	md5

Configurare il database per usare UTF-8.

Editare postgresql.conf e settare un non-zero value per max_prepared_transactions.



NB: The default value, 0, disables prepared transactions, which is not recommended for Bonita BPM Engine. The value should be at least as large as the value set for max_connections (default is typically 100). See the PostgreSQL documentation for details.

Inizializzazione e setup

Dopo aver installato PostgreSQL e PostgreSQL-server procediamo con l'inizializzazione e setup con il seguente comando:

```
postgresql-setup --initdb
```

Dovremo ricevere il seguente output:

```
* Initializing database in '/var/lib/pgsql/data'
* Initialized, logs are in /var/lib/pgsql/initdb_postgresql.log
```

Ora abilitiamo e facciamo partire il servizio PostgreSQL con systemctl

```
systemctl enable --now postgresql
```

Creazione utente e db

Dopo aver fatto login come utente postgres con il comando:

```
su - postgres
```

Lanciamo il seguente comando:

```
psql
```

Dopodiché aggiungiamo l'utente bonita e creiamo i due database necessari al corretto funzionamento;

```
postgres=# CREATE USER bonita WITH PASSWORD 'bonita';
postgres=# CREATE DATABASE bonita7 WITH ENCODING 'UTF8' OWNER bonita;
postgres=# CREATE DATABASE bonita7_bd WITH ENCODING 'UTF8' OWNER bonita;
```



```
postgres=# \q
```

Scegli il database

Nella cartella bin editare il file setenv.sh e settare come segue:

```
...
# Define the RDMBS vendor use by Bonita Engine to store data
DB_OPTS="-Dsysprop.bonita.db.vendor=postgres"

# Define the RDMBS vendor use by Bonita Engine to store Business Data
# If you use different DB engines by tenants, please update directly bonita-tenant-community-
custom.properties
BDM_DB_OPTS="-Dsysprop.bonita.bdm.db.vendor=postgres"
...
```

Driver JDBC

Scarichiamo i Driver JDBC per postgres [qui](#) e mettiamoli nella cartella /opt/bonita7/lib/bonita .

Configurazione Data Source

Editare il file /conf/bitronix-resources.properties e una volta commentate le righe che fanno riferimento al db H2 configurare correttamente quelle relative a postgres. Esempio:

```
...
# PostgreSQL example
resource.ds1.uniqueName=jdbc/bonitaDSXA
resource.ds1.minPoolSize=1
resource.ds1.maxPoolSize=50

resource.ds1.className=org.postgresql.xa.PGXADatasource
resource.ds1.driverProperties.user=bonita
resource.ds1.driverProperties.password=bonita
resource.ds1.driverProperties.serverName=localhost
resource.ds1.driverProperties.portNumber=5432
resource.ds1.driverProperties.databaseName=bonita7
resource.ds1.testQuery=SELECT 1
...
# PostgreSQL example
resource.ds2.uniqueName=jdbc/BusinessDataDSXA
resource.ds2.minPoolSize=0
resource.ds2.maxPoolSize=5

resource.ds2.className=org.postgresql.xa.PGXADatasource
resource.ds2.driverProperties.user=bonita
resource.ds2.driverProperties.password=bonita
resource.ds2.driverProperties.serverName=localhost
resource.ds2.driverProperties.portNumber=5432
resource.ds2.driverProperties.databaseName=bonita7_bd
resource.ds2.testQuery=SELECT 1
...
```

Editare il file conf/Catalina/localhost/bonita.xml e agire come prima, avendo cura di commentare le risorse già presenti come segue:

```
<!-- ##### -->
<!-- Configure Datasource for Bonita BPM standard database -->
<!-- ##### -->
...
<!--
  <!-- H2 Example -->
  <Resource name="bonitaSequenceManagerDS"
    auth="Container"
    type="javax.sql.DataSource"
    maxActive="17"
    minIdle="5"
    maxWait="10000"
```



```

        initialSize="3"
        validationQuery="SELECT 1"
        validationInterval="30000"
        removeAbandoned="true"
        logAbandoned="true"
        username="sa"
        password=""
        driverClassName="org.h2.Driver"
url="jdbc:h2:tcp://localhost:9091/bonita_journal.db;MVCC=TRUE;DB_CLOSE_ON_EXIT=TRUE;IGNORECASE=TRUE;"/>
-->
...
<!-- ##### -->
<!-- Configure Business Data Datasource -->
<!-- ##### -->
...
<!--
<Resource name="NotManagedBizDataDS"
    auth="Container"
    type="javax.sql.DataSource"
    maxActive="17"
    minIdle="5"
    maxWait="10000"
    initialSize="3"
    validationQuery="SELECT 1"
    validationInterval="30000"
    removeAbandoned="true"
    logAbandoned="true"
    username="sa"
    password=""
    driverClassName="org.h2.Driver"
url="jdbc:h2:tcp://localhost:9091/business_data.db;MVCC=TRUE;DB_CLOSE_ON_EXIT=TRUE;IGNORECASE=TRUE;"/>
-->

```

Esempio:

```

<!-- ##### -->
<!-- Configure Datasource for Bonita BPM standard database -->
<!-- ##### -->
...
<!-- Postgresql Example -->
    <Resource name="bonitaSequenceManagerDS"
        auth="Container"
        type="javax.sql.DataSource"
        maxActive="17"
        minIdle="5"
        maxWait="10000"
        initialSize="3"
        validationQuery="SELECT 1"
        validationInterval="30000"
        removeAbandoned="true"
        logAbandoned="true"
        username="bonita"
        password="bonita"
        driverClassName="org.postgresql.Driver"
        url="jdbc:postgresql://localhost:5432/bonita7"/>
...
<!-- ##### -->
<!-- Configure Business Data Datasource -->
<!-- ##### -->
...
<!-- Postgresql Example -->
    <Resource name="NotManagedBizDataDS"
        auth="Container"

```



```

type="javax.sql.DataSource"
maxActive="17"
minIdle="5"
maxWait="10000"
initialSize="3"
validationQuery="SELECT 1"
validationInterval="30000"
removeAbandoned="true"
logAbandoned="true"
username="bonita"
password="bonita"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://localhost:5432/bonita7_bd"/>
...

```

Rimuoviamo il listener h2 (così che h2 non venga lanciato in automatico), commentando la linea del listener h2 nel file: `/opt/bonita7/conf/server.xml`

```

<!-- Listener className="org.bonitasoft.tomcat.H2Listener" tcpPort="9091"
baseDir="${catalina.home}/bonita/engine-server/work/platform" start="true" /-->

```

E cambiamo anche le porte di default scendendo di 1000 (es. 8080 diventerà 7080).



In caso di importazioni corpose bisogna settare la property di timeout correttamente in `conf/bitronix-config.properties`

```
bitronix.tm.timer.defaultTransactionTimeout=180
```

Nel file `setenv(bat|sh)` possiamo inserire i valori corretti di sorgenti e risorse(es: Ram destinata al tomcat) <NB:Esempio per LINUX>

```

#!/bin/sh

# Set some JVM system properties required by Bonita BPM

# Bonita home folder (configuration files, temporary folder...) location
BONITA_HOME="-Dbonita.home=${CATALINA_HOME}/bonita"

# Define the RDMBS vendor use by Bonita Engine to store data
DB_OPTS="-Dsysprop.bonita.db.vendor=postgres"

# Define the RDMBS vendor use by Bonita Engine to store Business Data
# If you use different DB engines by tenants, please update directly bonita-tenant-community-
custom.properties
BDM_DB_OPTS="-Dsysprop.bonita.bdm.db.vendor=postgres"

# Bitronix (JTA service added to Tomcat and required by Bonita Engine for transaction management)
BTM_OPTS="-Dbtm.root=${CATALINA_HOME} -Dbitronix.tm.configuration=${CATALINA_HOME}/conf/bitronix-
config.properties"

# Optionnal JAAS configuration. Usually used when delegating authentication to LDAP / Active
Directory server
#SECURITY_OPTS="-Djava.security.auth.login.config=${CATALINA_HOME}/conf/jaas-standard.cfg"

# Pass the JVM system properties to Tomcat JVM using CATALINA_OPTS variable
CATALINA_OPTS="${CATALINA_OPTS} ${BONITA_HOME} ${DB_OPTS} ${BDM_DB_OPTS} ${BTM_OPTS} -
Dfile.encoding=UTF-8 -Xshare:auto -Xms1536m -Xmx1536m -XX:MaxPermSize=256m -
XX:+HeapDumpOnOutOfMemoryError"
export CATALINA_OPTS

# Only set CATALINA_PID if not already set (check for empty value) by startup script (usually done
by /etc/init.d/tomcat7 but not by startup.sh nor catalina.sh)
if [ -z ${CATALINA_PID+x} ]; then
    CATALINA_PID=${CATALINA_BASE}/catalina.pid;
    export CATALINA_PID;

```



Una volta avviata l'applicazione bisogna collegarsi con l'utente install\install e creare un utente administrator, i gruppi ed i ruoli necessari.

In caso si volesse partire con un db pulito già pronto (in caso di problemi con l'installazione) questa è la procedura partendo dai file di dump che sono scaricabili da:

smb://storage.bo.priv/RESOURCES/Bonita/bonita7_dump/

1. Drop database come utente postgres;
2. Creazione database vuoto come utente postgres;
3. cat bonita7.sql |psql -U bonita -d bonita7 -h 10.17.61.242;
4. cat bonita7_bd.sql |psql -U bonita -d bonita7_vd -h 10.17.61.242;



Accedendo alla pagina web di Bonita, per il primo accesso bisogna utilizzare il nome utente *"install"* con la password *"install"*.

Successivamente:

1. creare l'utente **"bonita"**;
2. in *"Organization"* -> *"Profiles"*, creare il profilo **"Users"**;
3. in *"Organization"* -> *"Roles"*, creare il ruolo **"membro"**
4. in *"Organization"* -> *"Groups"*, creare i gruppi **"TUTTI"** ed **"HELP"** [importante lo stampatello maiuscolo]

Windows

In caso di installazione in produzione sarebbe il caso di installare un tomcat7 pulito e gestirlo come servizio. Ho eseguito dei test di installazione in una macchina windows e sono riuscito a far partire correttamente bonita 7 su postgresQL con le seguenti properties:

```
-Dcatalina.home=E:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7Bonita7
-Dcatalina.base=E:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7Bonita7
-Djava.endorsed.dirs=E:\Program Files\Apache Software Foundation\Tomcat
7.0_Tomcat7Bonita7\endorsed
-Djava.io.tmpdir=E:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7Bonita7\temp
-Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager
-Djava.util.logging.config.file=E:\Program Files\Apache Software Foundation\Tomcat
7.0_Tomcat7Bonita7\conf\logging.properties
-Dbonita.home=E:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7Bonita7\bonita
-Dsysprop.bonita.db.vendor=postgres
-Dsysprop.bonita.bdm.db.vendor=postgres
-Dbtm.root=E:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7Bonita7
-Dbitronix.tm.configuration=E:\Program Files\Apache Software Foundation\Tomcat
7.0_Tomcat7Bonita7\conf\bitronix-config.properties
-Djava.security.auth.login.config=E:\Program Files\Apache Software Foundation\Tomcat
7.0_Tomcat7Bonita7\conf\jaas-standard.cfg
-Dfile.encoding=UTF-8
-Xshare:auto
```

Installare il Tomcat7 0.55 come servizio (nel caso di installazione su server dove la porta 8080 risulti già in uso utilizzare ad esempio la porta 9080) Dopo aver scaricato il bundle di deploy in Tomcat7 modificare il file conf/server.xml ad hoc e di seguito i file sopra indicati. Ricordarsi di inserire le variabili all'interno dell'eseguibile tomcat7w.exe del Tomcat 7 appena installato.

Installazione Bonita7View

1. Unzippare il .war in una cartella temporanea e poi spostarla nella directory /opt/tomcat/webapps;
2. Configurare il file *bonitaView.properties* in /opt/tomcat7/webapps/bonita7View/WEB-INF/classes, inserendo la URL corretta del Bonita a cui fa riferimento, insieme al nome utente e la password di quest'ultimo:

```
# paramteri di connessione alla webApp di bonita tramite API
serviceURL = http://localhost:7080    ## "localhost" se bonita è presente nella stessa macchina
applicationName = bonita
```



```
username = bonita          ## username e password che sono stati impostati su bonita
password = bonita

# formato di data e ora da utilizzare per visualizzare le info sui task (default: dd/MM/yyyy
HH:mm:ss)
dateFormat = dd/MM/yyyy HH:mm

# numero massimo di risultati resitutiti dalle query tramite API di bonita
maxResult = 10000

# bordo presente attorno l'immagine esportata nel .bar
imageBorderX = 10
imageBorderY = 10
```

3. Ricordarsi di assegnare la directory di Bonita7View allo stesso utente a cui è stato assegnato tomcat;
4. Fare un restart del servizio di Tomcat.