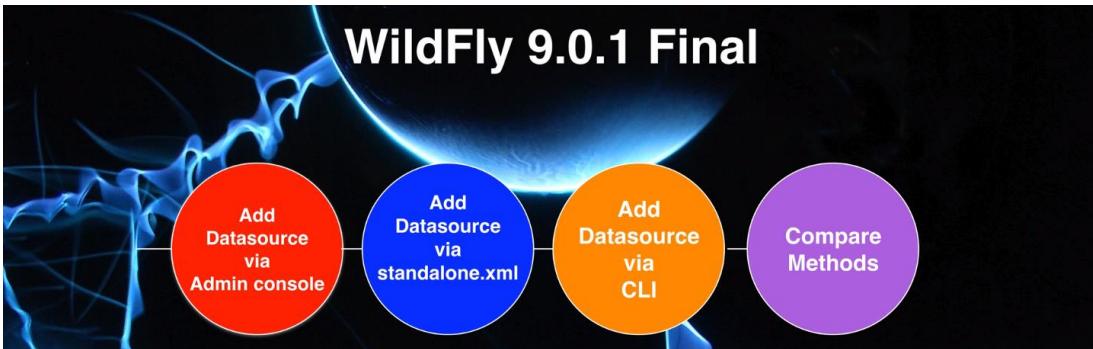


Three Ways to Add a Datasource to Wildfly 9 sshots



The goal is to go from a fresh wildfly install to adding a datasource



Start up your Wildfly server, we are going to be using PostgreSQL as the datasource. We need to also start PostgreSQL and create our database in it.

File Explorer windows showing the directory structure of Wildfly-23.0.2.Final/bin:

```

    Elite8300 > My Documents > Wildfly > wildfly-23.0.2.Final > bin
    Elite8300 > My Documents > Wildfly > wildfly-23.0.2.Final > bin

```

Name	Date modified	Type
client	4/27/2021 7:25 AM	File folder
JBOSSCLRC File	4/27/2021 7:25 AM	
add-user	4/27/2021 7:25 AM	Windows Batch File
add-user.properties	4/27/2021 7:25 AM	PROPERTIES File
add-user	4/27/2021 7:25 AM	PSI File
add-user	4/27/2021 7:25 AM	SH File
appclient	4/27/2021 7:25 AM	Windows Batch File
appclient.conf	4/27/2021 7:25 AM	CONF File
appclient.conf	4/27/2021 7:25 AM	Windows Batch File
appclient.conf	4/27/2021 7:25 AM	PSI File
appclient	4/27/2021 7:25 AM	PSI File
appclient	4/27/2021 7:25 AM	SH File
common	4/27/2021 7:25 AM	Windows Batch File
common	4/27/2021 7:25 AM	PSI File
common	4/27/2021 7:25 AM	SH File
domain	4/27/2021 7:25 AM	Windows Batch File

Terminal windows showing the execution of standalone.sh and standalone.bat scripts:

```

MINGW64/c/Users/Elite8300/Documents/Wildfly/wildfly-23.0.2.Final/bin
14:34:41,249 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on https://127.0.0.1:9990
5/11/2021 5:17 PM
$ ./standalone.sh
=====
JBoss Bootstrap Environment
JBoss_HOME: /c/Users/Elite8300/Documents/Wildfly/wildfly-23.0.2.Final
JAVA_OPTS: -server Xms64m >Xm128m >>Xmx896m >>Xms896m
e=256m -Djava.net.preferIPv4Stack=true -Dboss.modules.system.plugins.org.jboss.byteman -Djava.awt.headless=true --add-exports=java.base/sun.nio.ch=ALL-UNNAMED --add-exports=jdk.unsupported/sun.unfactored=ALL-UNNAMED
JAVA_OPTS: -server Xms64m >Xm128m >>Xmx896m >>Xms896m
e=256m -Djava.net.preferIPv4Stack=true -Dboss.modules.system.plugins.org.jboss.byteman -Djava.awt.headless=true --add-exports=java.base/sun.nio.ch=ALL-UNNAMED --add-exports=jdk.unsupported/sun.unfactored=ALL-UNNAMED
5/11/2021 5:17 PM
Unable to read the logging configuration from file:/c/Users/Elite8300/Documents/Wildfly/wildfly-23.0.2.Final/standalone/configuration/logging.properties (The system cannot find the path specified)
53 PM

04:15:55,168 INFO [org.infinispan.CONFIG] (MSC service thread 1-4) ISPN000152: Infinispan 11.0.10 final (build:git-1333-1a2a2d5 date:20201208-14:25:42) starting up
04:15:58,421 INFO [org.infinispan.PERSISTENCE] (ServerService Thread Pool -- 79) ISPN000001: Persistence layer will be passivated
04:15:58,426 INFO [org.infinispan.marshall] (ServerService Thread Pool -- 79) ISPN000002: Starting user marshaller org.wildfly.clustering.infinispan.spi.marshalling.InfinispanMarshallerParallel
04:16:06,706 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 79) WFLYCLINF0002: Started http-remoting-connector cache from ejb container
04:16:15,696 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool - 84) RESTEAS00225: Deploying javax.ws.rs.core.Application: class com.wildfly.extension.undertow.WildFlyExtension
04:16:15,696 INFO [org.wildfly.extension.undertow] (ServerService Thread Pool - 84) WFLYUT0021: Registered web context: '/story' for server 'default-server'
04:16:16,539 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed 'story-0.0.1-SNAPSHOT.war' (runtime-name : 'story-0.0.1-SNAPSHOT.war')
04:16:16,550 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed 'story-0.0.1-SNAPSHOT.war' (runtime-name : 'story-0.0.1-SNAPSHOT.war')
04:16:16,550 INFO [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0010: Deployed 'story-0.0.1-SNAPSHOT.war' (runtime-name : 'story-0.0.1-SNAPSHOT.war')
04:16:16,731 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: Wildfly management interface started on http://127.0.0.1:9990/management
04:16:16,731 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://127.0.0.1:9990/management
04:16:16,731 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9990
53 PM

```



## HAL Management Console

### Deployments

- Deploy an Application | Start
- Deploy an application to the server
- 1. Use the 'Add Deployment' wizard to deploy the application
- 2. Enable the deployment

### Runtime

- Monitor the Server | Start
- View runtime information such as server status, JVM status, and server log files.
- Failure

localhost:9990/console/index.html#deployments

HAL Management Console

Homepage Deployments Configuration Runtime Patching Access Control

**Deployment**

Filter by name or deployment status

- postgresql-42.2.22.jar
- story-0.0.1-SNAPSHOT.war

**Deployments**

A deployment represents anything that can be deployed (e.g. an application such as EJB-JAR, WAR, EAR, any kind of standard archive such as RAR or JBoss-specific deployment) into a server. You can use drag and drop to add new content or replace existing deployments. Simply drag one or several files onto the deployment column. If there's already a deployment with the same name, the deployment will be replaced, otherwise the deployment will be added. The drop will be enabled by default.

localhost:9990/console/index.html#runtime:path=standalone-server-column--standalone-host-elite8300-pcrss-datasources

HAL Management Console

Homepage Deployments Configuration Runtime Patching Access Control

**Server**

Monitor	Data source
elite8300-pc	ExampleDS

**Datasources**

Provides access to runtime operations such as 'Test Connection' and flush operations.

In order to view datasource and JDBC statistics, please make sure to enable them in the configuration section.

The screenshot shows a Java project structure on the left and a code editor on the right. The code editor displays a Docker Compose file:

```

version: '3'
services:
  postgres:
    image: postgres
    container_name: pgdocker
    ports:
      - 5432:5432
    restart: always
    environment:
      POSTGRES_USER: "postgres"
      POSTGRES_PASSWORD: "postgres"
  padmin4:
    image: 'dpage/pgadmin4:latest'
    restart: always
    ports:
      - 8081:80
    depends_on:
      - postgres
    environment:
      PGADMIN_DEFAULT_EMAIL: "teejay@gmail.com"
      PGADMIN_DEFAULT_PASSWORD: "123456"
      PGADMIN_CONFIG_SERVER_NODE: "False"

```

The screenshot shows two terminal windows. The left window shows the command `docker-compose up` being run, and the right window shows the logs of the PostgreSQL and PgAdmin containers.

```

elite8300@elite8300-PC MINGW64 ~/Documents/STS4Workspace/story
$ docker-compose up
Creating network "story_default" with the default driver
Creating pgdocker ... done
Creating story_padmin4_1 ... done
Attaching to pgdocker, story_padmin4_1
pgdocker  | The files belonging to this database system will be owned by user
pgdocker  | "postgres". This user must also own the server process.
pgdocker  | The database cluster will be initialized with locale "en_US.utf8".
pgdocker  | The default database encoding has accordingly been set to "UTF8".
pgdocker  | The default text search configuration will be set to "english".
pgdocker  | Data page checksums are disabled.
pgdocker  | fixing permissions on existing directory /var/lib/postgresql/data ...
pgdocker  | creating subdirectories ... ok
pgdocker  | selecting dynamic shared memory implementation ... posix
pgdocker  | selecting default max_connections ... 100
pgdocker  | selecting default shared_buffers ... 128MB
pgdocker  | selecting default time zone ... Etc/UTC
pgdocker  | creating configuration files ... ok
pgdocker  | running bootstrap script ... ok

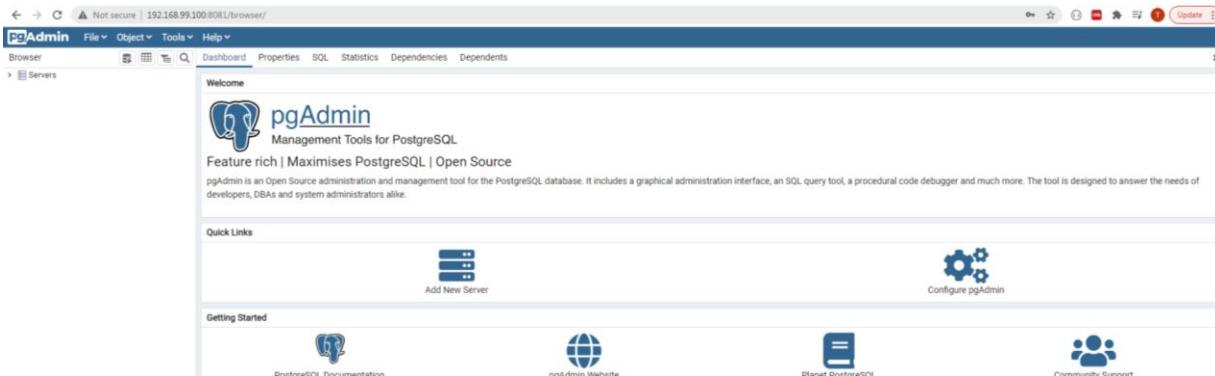
```

```

syncing data to disk ... ok
pgdocker  | initdb: warning: enabling "trust" authentication for local connections
pgdocker  | You can change this by editing pg_hba.conf or using the option --auth-local or --auth-host.
pgdocker  | --auth-local and --auth-host, the next time you run initdb.
Success. You can now start the database server using:
      pg_ctl -D /var/lib/postgresql/data -l logfile start
LOG:  starting PostgreSQL 13.3 (Debian 13.3-1.pgdg100+1) on x86_64-pc-linux-gnu,
       compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
[2021-06-18 22:51:51.372 UTC] [45] LOG:  listening on Unix socket
[2021-06-18 22:51:51.377 UTC] [46] LOG:  database system was shutdown at 2021-06-18 22:51:51.150 UTC
[2021-06-18 22:51:51.384 UTC] [45] LOG:  database system is ready to accept connections
pgdocker  | done
pgdocker  | server started
/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/
pgdocker  | 
pgdocker  | waiting for server to start...2021-06-18 22:51:51.370 UTC [45]
pgdocker  | [2021-06-18 22:51:51.372 UTC] [45] LOG:  listening on Unix socket
[2021-06-18 22:51:51.377 UTC] [46] LOG:  database system was shutdown at 2021-06-18 22:51:51.150 UTC
[2021-06-18 22:51:51.384 UTC] [45] LOG:  database system is ready to accept connections
pgdocker  | done
pgdocker  | server started
/usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/
[2021-06-18 22:51:51.564 UTC] [45] LOG:  received fast shutdown request
pgdocker  | waiting for server to shut down...2021-06-18 22:51:51.566 UTC [45]
[5] LOG:  aborting any active transactions
[2021-06-18 22:51:51.576 UTC] [45] LOG:  background worker "logical replicator" (PID 52) exited with exit code 1
[2021-06-18 22:51:51.578 UTC] [47] LOG:  shutting down
[2021-06-18 22:51:51.592 UTC] [45] LOG:  database system is shutdown
pgdocker  | done
pgdocker  | server stopped
PostgreSQL init process complete; ready for start up.
pgdocker  | 
pgdocker  | PostgreSQL init process complete; ready for start up.
[2021-06-18 22:51:51.720 UTC] [1] LOG:  starting PostgreSQL 13.3 (Debian 13.3-1.pgdg100+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 8.3.0-6) 8.3.0, 64-bit
pgdocker  | [2021-06-18 22:51:51.721 UTC] [1] LOG:  listening on IPv4 address "0.0.0.0", port 5432
pgdocker  | [2021-06-18 22:51:51.721 UTC] [1] LOG:  listening on IPv6 address "::", port 5432
pgdocker  | [2021-06-18 22:51:51.723 UTC] [1] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
[2021-06-18 22:51:51.727 UTC] [64] LOG:  database system was shutdown at 2021-06-18 22:51:51.151 UTC
pgadmin4_1  | NOTE: Configuring authentication for DESKTOP mode.
pgadmin4_1  | [2021-06-18 22:52:09 +0000] [1] [INFO] Starting gunicorn 20.1.0
pgadmin4_1  | [2021-06-18 22:52:09 +0000] [1] [INFO] Listening at: http://[::]:80 (1)
pgadmin4_1  | [2021-06-18 22:52:09 +0000] [1] [INFO] Using worker: gthread

```

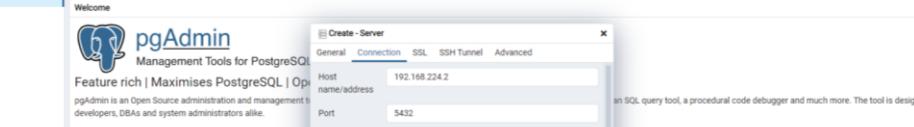
Start the creation of the PostgreSQL and PgAdmin using the docker-compose file



Open PgAdmin on localhost:9990 console and log in with the password

The screenshot shows the pgAdmin 4 interface. A context menu is open over a 'Servers' entry in the left sidebar, with 'Create' selected. A sub-menu titled 'Server...' is displayed, containing options: 'Refresh...', 'Remove Server Group', and 'Properties...'. The main workspace displays the pgAdmin logo and the tagline 'Management Tools for PostgreSQL'. Below the logo, it says 'Feature rich | Maximises PostgreSQL | Open Source'. A note at the bottom states: 'pgAdmin is an Open Source administration and management tool for the PostgreSQL database. It includes a graphical administration interface, an SQL query tool, a procedural code debugger and much more. The tool is designed to answer the needs of'. The title bar shows the URL 'Not secure 192.168.99.100:8081/browser/' and the pgAdmin logo.

Use the `docker inspect pgdocker` command to get the `IPAddress` value to use to connect to the PostgreSQL 13.3-1



The screenshot shows the pgAdmin interface with a 'Create - Server' dialog box open. The dialog has tabs for General, Connection, SSL, SSH Tunnel, and Advanced. The 'Connection' tab is selected, showing fields for Host (192.168.224.2), Port (5432), Maintenance database (postgres), Username (postgres), Kerberos authentication (False), Password (redacted), Save password (unchecked), Role (empty), and Service (empty). Below the dialog are 'Quick Links' and 'Getting Started' sections, and a sidebar with 'Servers' selected.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays a tree view of the database structure under the 'Servers' node, specifically for the 'mpgdb' database. The visible nodes include 'Databases', 'Schemas', 'Tables', 'Views', 'Functions', 'Procedures', 'Triggers', 'Sequences', 'Extensions', 'Languages', 'Publications', and 'Tablespaces'. The main pane is divided into two sections: 'Database sessions' and 'Transactions per second'. The 'Database sessions' section shows a table with columns: 'Tuples in' (with values 1 and 0), 'Tuples out' (with values 30, 25, 20, 15, 10, 5, 0), and 'Block I/O' (with values 60, 50, 40, 30, 20, 10, 0). The 'Transactions per second' section shows a chart with values 3, 2, 1, and 0. Below these sections is a 'Server activity' tab with tabs for 'Sessions', 'Locks', and 'Prepared Transactions'. A search bar and a refresh button are also present.

There are no tables at the moment inside our mypgdb PostgreSQL database

## 1 Add JDBC Driver



The 2 steps are to load the JDBC driver into Wildfly, then configure the datasource

Postgres JDBC Driver

Search this to go to the download page for the JDBC driver

**jdbc:postgresql://HOST:PORT/DATABASE**

Host:Port                      Database  
**jdbc:postgresql://200.100.50.10:5432/demo1**

This is the JDBS connection string for your database driver like above

```
bin - java - 80x24
Daniels-MacBook-Pro:bin dspiess$ ./standalone.sh
=====
JBoss Bootstrap Environment
JBoss_HOME: /Users/dspiess/java/wildfly/wildfly-9.0.1.Final
JAVA: /Library/Java/JavaVirtualMachines/jdk1.8.0_25.jdk/Contents/Home/bin/java
JAVA_OPTS: -server -XX:+UseCompressedOops -server -XX:+UseCompressedOops -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true
=====
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option MaxPermSize=256m; support was removed in 8.0
21:22:00,457 INFO [org.jboss.modules] (main) JBoss Modules version 1.4.3.Final
21:22:01,192 INFO [org.jboss.msc] (main) JBoss MSC version 1.2.6.Final
21:22:01,244 INFO [org.jboss.as] (MSC service thread 1-6) WFLYSRV0049: WildFly Full 9.0.1.Final (WildFly Core 1.0.1.Final) starting
```

Pick your JDBC driver that you downloaded

The screenshots illustrate the deployment process in WildFly:

- Step 1: Upload Deployment**  
A file named "postgresql-9...01.jdbc4.jar" is selected for upload.
- Step 2: Verify Upload**  
The deployment is verified with the following settings:
  - Name:** postgresql
  - Runtime Name:** postgresql-9.4-1201.jdbc4.jar
  - Enable:** checked
- Step 3: Deployment List**  
The deployment "postgresql" is listed in the deployment list.

The **Name** is how the deployment is known to your user, the **Runtime Name** is how it is known to the server. This allows you to have multiple Names that point to the same Runtime Name.

The deployment "postgresql" is now listed in the deployment list under the "Deployment" section.

Your JDBC driver is now available

The "Create a datasource" wizard is open, showing the following steps:

- Add a new datasource and follow the steps in the Create Datasource wizard.
- > [Datasources](#)

The "Datasources" link is circled in blue.

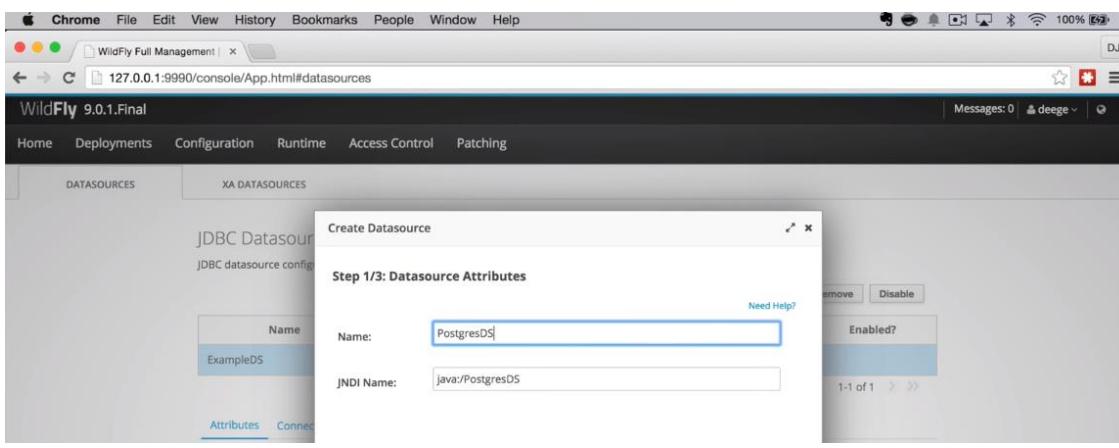
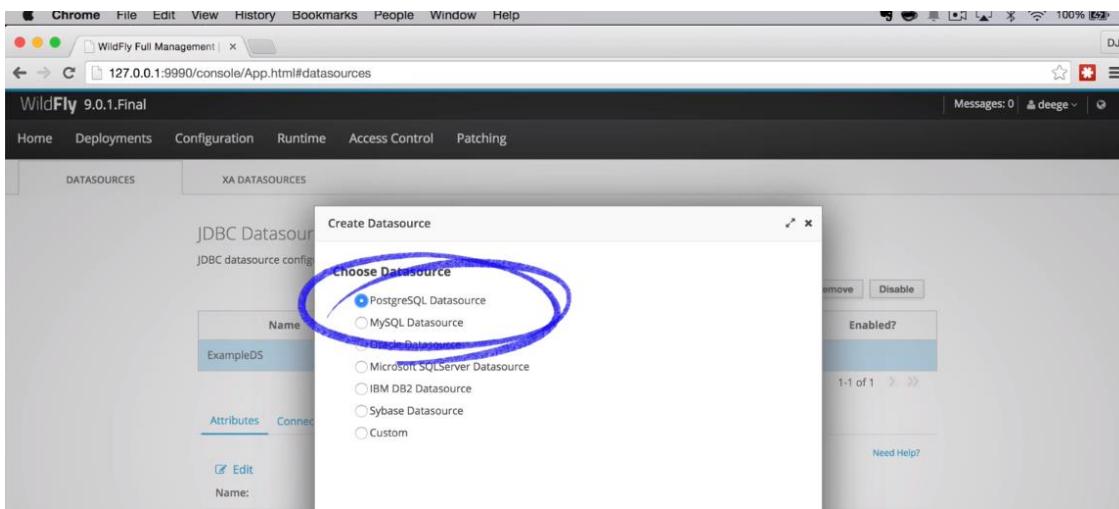
**Left sidebar (Common Tasks):**

- Deploy an application**: To deploy an application.
  1. Add a deployment.
  2. Enable the new deployment.[Create Deployment](#)
- Apply a patch**: Update to the latest patch level. The patch file must be downloaded to [link]

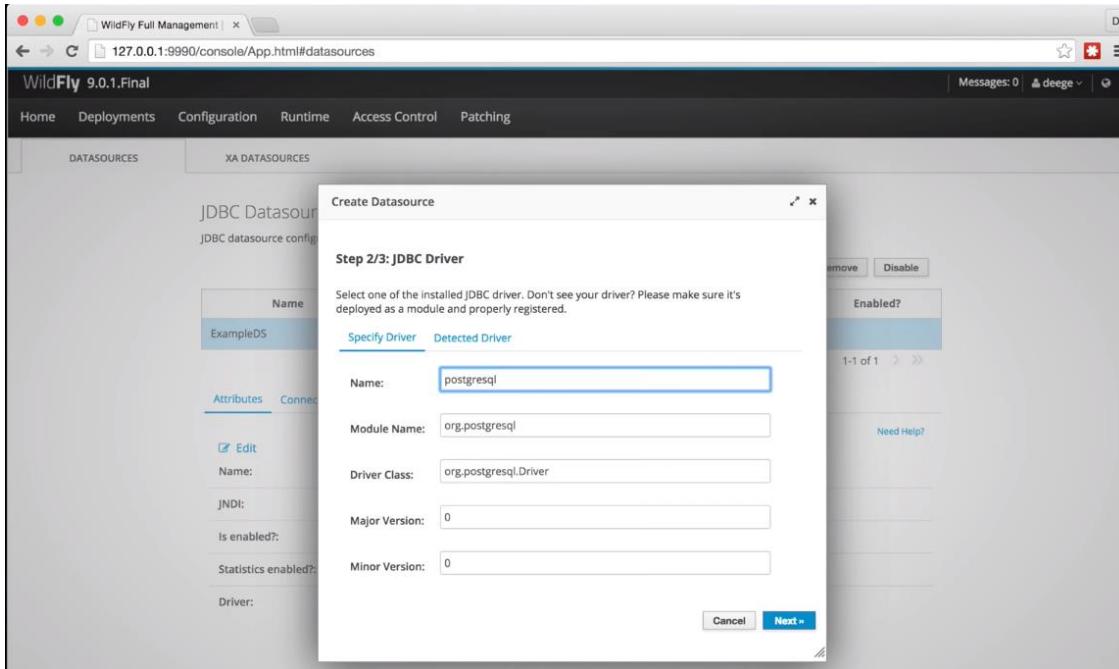
**Right sidebar (Find More Resources):**

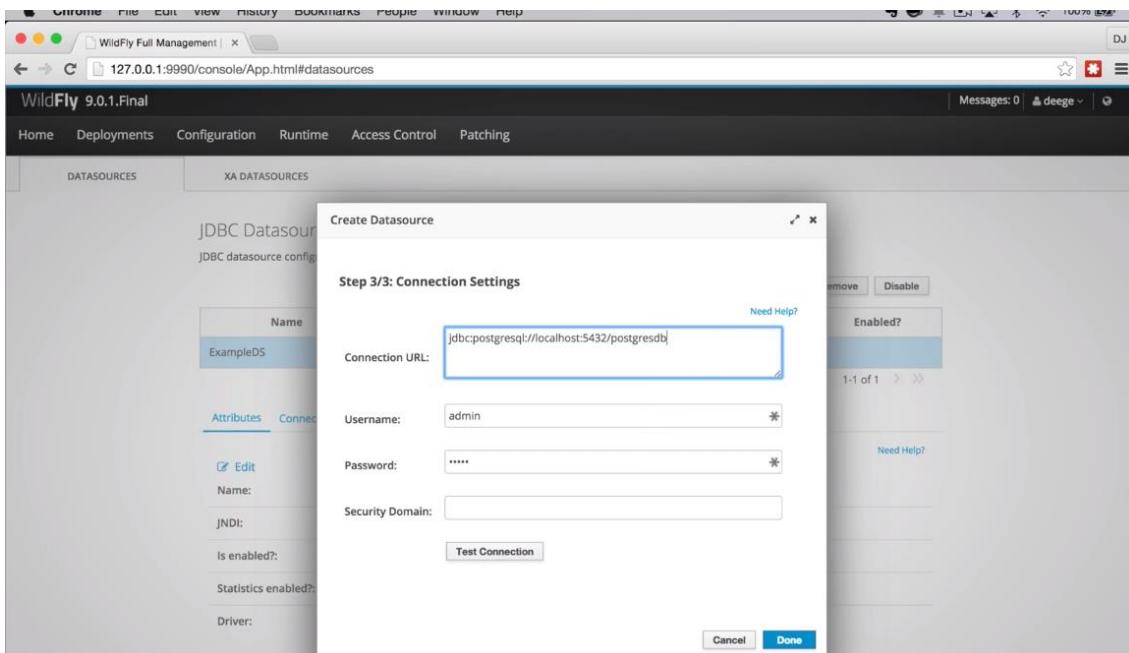
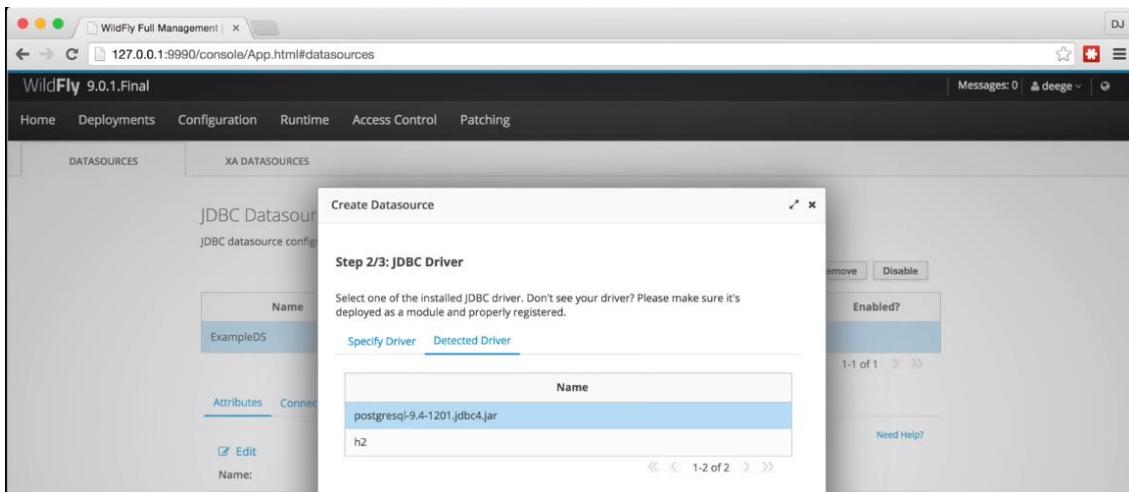
- General Resources
  - WildFly Home
  - WildFly Documentation
  - Admin Guide
  - Model Reference Documentation
  - Browse Issues
  - Latest News
- Get Help
  - Access tutorials and quickstarts
  - User Forums
  - IRC
  - Developers Mailing List

Our next task is to create the Datasource

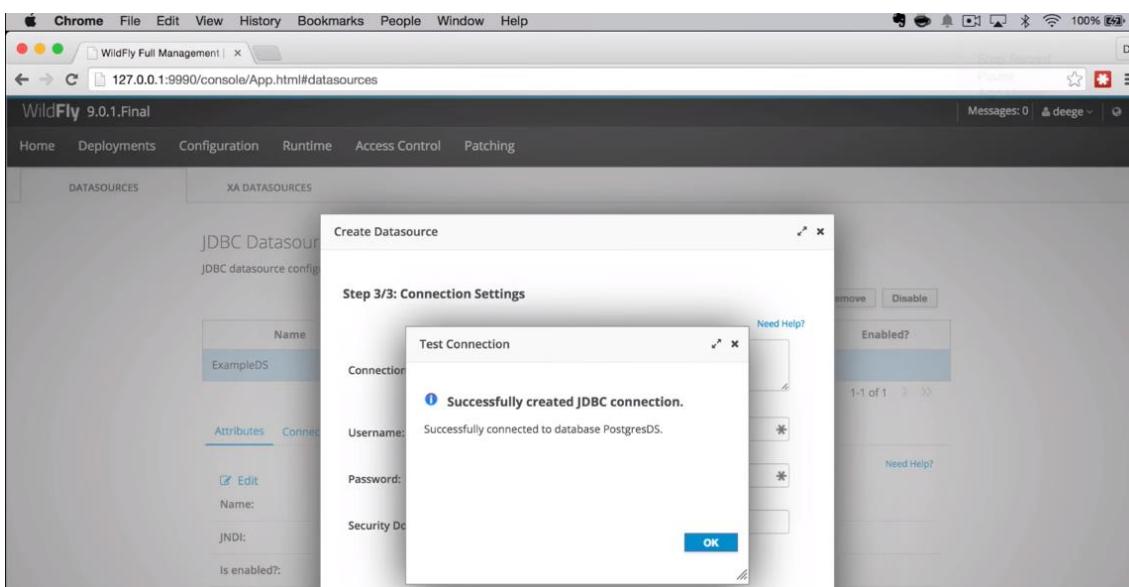


This is how your application is going to see your datasource, this must be unique





Update the connection URI to match your server, and update the username and password to match your credentials



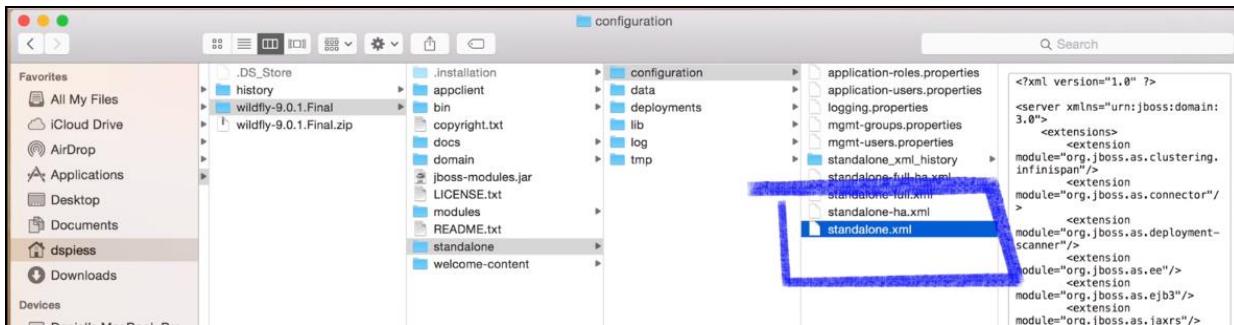
Test the connection

The screenshot shows the WildFly 9.0.1.Final management console. In the top navigation bar, 'Configuration' is selected. Below it, the 'DATASTORES' tab is active. A green banner at the top right says 'Added Datasource PostgresDS'. The main content area is titled 'JDBC Datasources' and shows a table of configurations. The table has columns for 'Name', 'JNDI', and 'Enabled?'. It contains two rows: 'ExampleDS' with JNDI 'java:jboss/datasources/ExampleDS' and 'PostgresDS' with JNDI 'java:/PostgresDS'. Both rows have a checked 'Enabled?' checkbox. Below the table are tabs for 'Attributes', 'Connection', 'Pool', 'Security', 'Properties', 'Validation', 'Timeouts', and 'Statements'. Under the 'Attributes' tab, there are fields for 'Name' (ExampleDS), 'JNDI' (java:jboss/datasources/ExampleDS), 'Is enabled?' (true), 'Statistics enabled?' (false), and 'Driver' (h2). There is also a link 'Need Help?'.

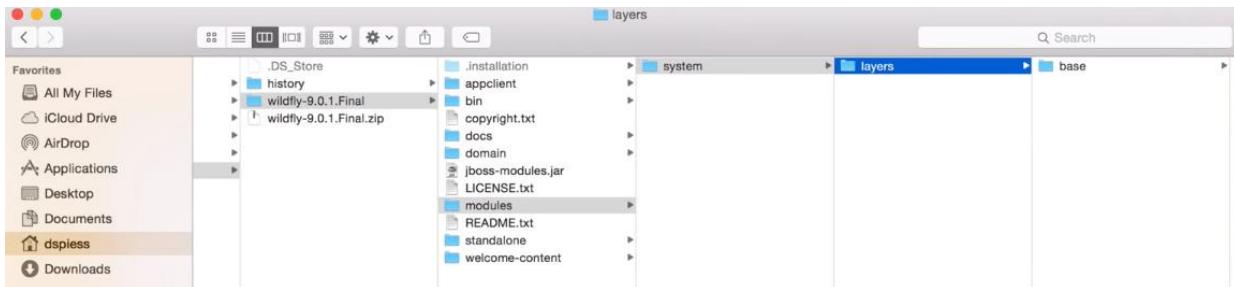
You have successfully added the datasource to Wildfly.



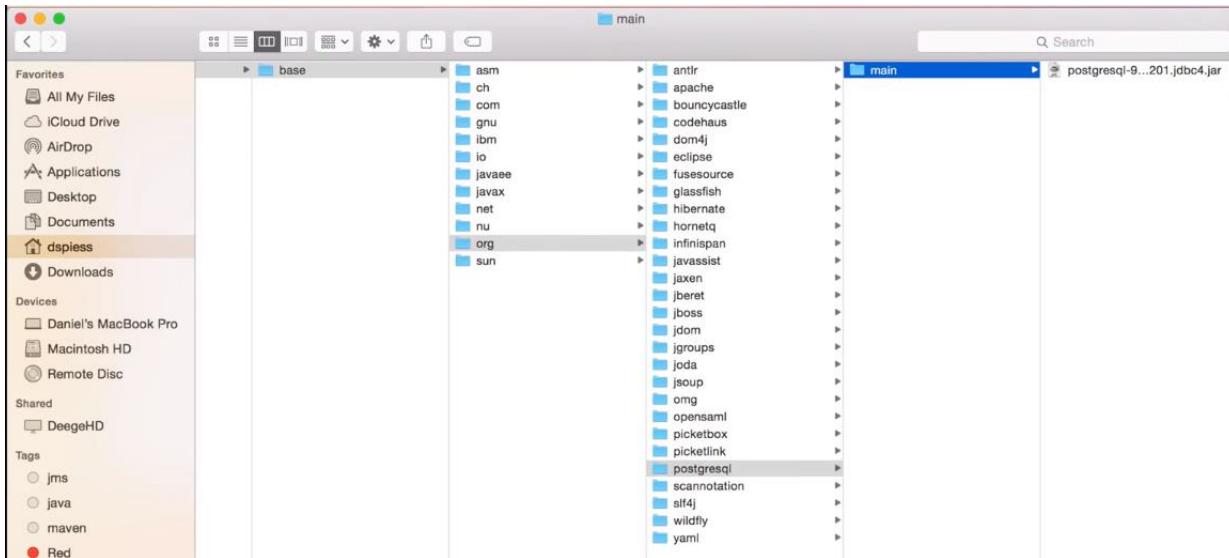
Let us now see how to add the PostgreSQL datasource to the Wildfly Application Server by editing the standalone.xml file within the Wildfly folder



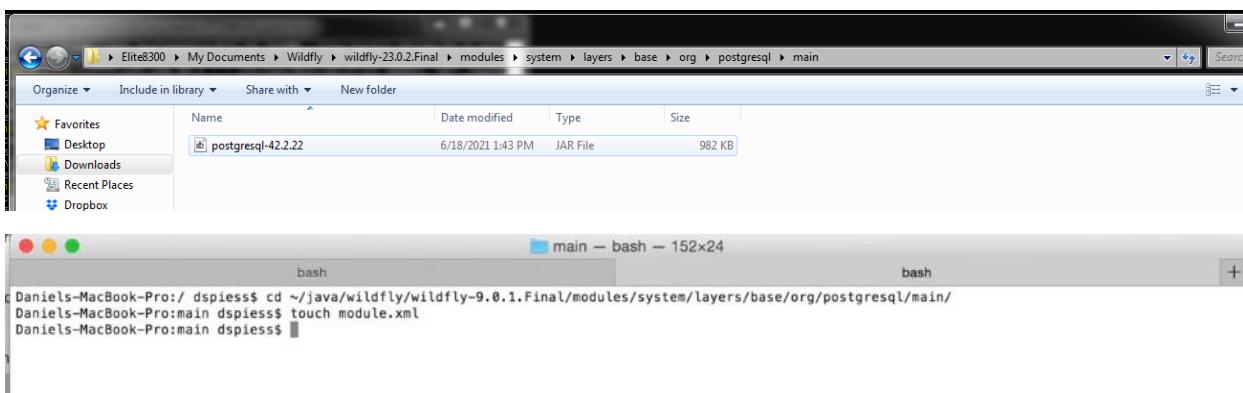
Behind the scene, the Wildfly management console is updating the standalone.xml file, the configuration script for your Wildfly server that contains configuration for everything you want to run in your server. Let us use manual approach.



First, we need to add the Postgres JDBC Driver manually



Then create the subfolders **org/postgresql/main**, this needs to match the package structure of your JDBC driver. Then copy your JDBC driver into the main folder location



You need to create an XML file in the **main** folder/directory called **module.xml**

### module.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<module xmlns="urn:jboss:module:1.1" name="org.postgresql">
  <resources>
    <resource-root path="postgresql-9.4-1201.jdbc4.jar"/>
  </resources>
  <dependencies>
    <module name="javax.api"/>
    <module name="javax.transaction.api"/>
  </dependencies>
</module>
```

File Explorer window showing the file structure:

```
C:\Users\Elite8300\Documents\Wildfly\wildfly-23.0.2.Final\modules\system\layers\base\org\main
```

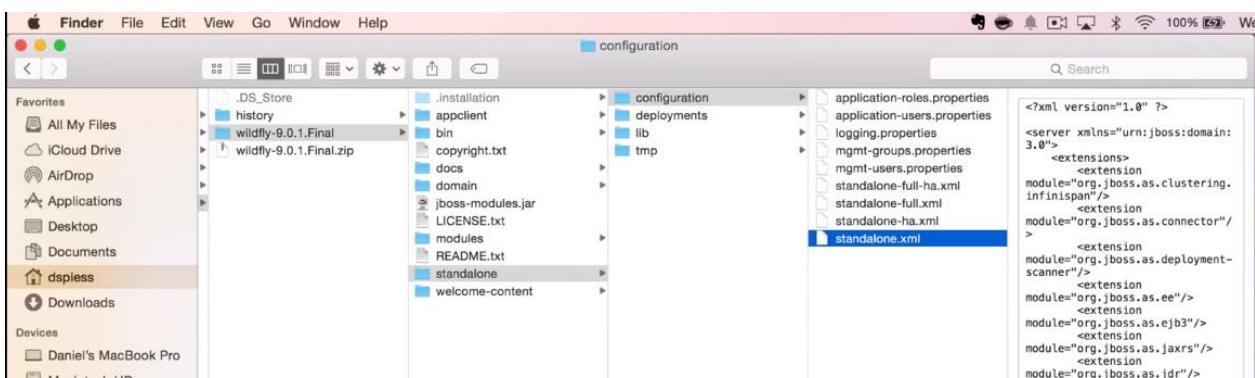
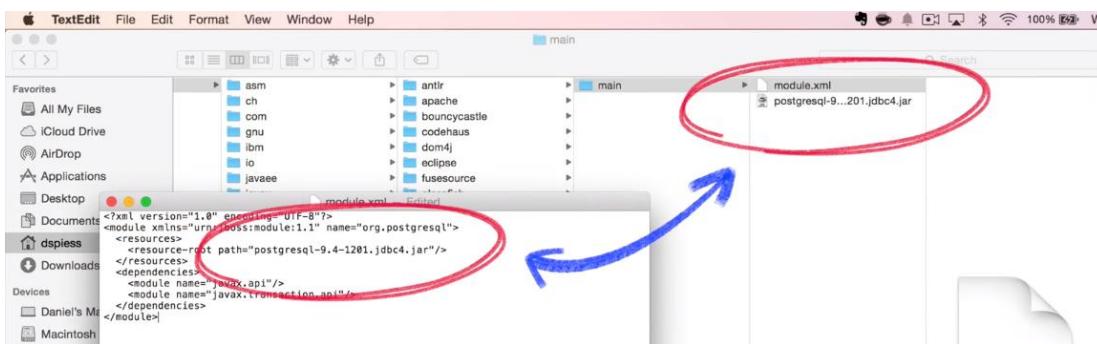
Name	Date modified	Type
module	6/21/2021 5:43 AM	XML Document
postgresql-42.2.22	6/18/2021 1:43 PM	JAR File

Code Editor window showing the module.xml file content:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <module xmlns="urn:jboss:module:1.1" name="org.postgresql">
3   <resources>
4     <resource-root path="postgresql-42.2.22.jar"/>
5   </resources>
6   <dependencies>
7     <module name="javax.api"/>
8     <module name="javax.transaction.api"/>
9   </dependencies>
10 </module>
```

File status bar: length : 303 li Ln : 1 Col : 1 Sel : 0 | 0 Dos\Windows UTF-8 INS



We now need to edit the standalone.xml file in the **Wildfly/standalone/configuration** folder/directory

```

<?xml version="1.0" ?>
<server xmlns="urn:jboss:domain:3.0">
    <extensions>
        <extension module="org.jboss.as.clustering.infinispan"/>
        <extension module="org.jboss.as.connector"/>
        <extension module="org.jboss.as.deployment-scanner"/>
        <extension module="org.jboss.as.ee"/>
        <extension module="org.jboss.as.ejb3"/>
        <extension module="org.jboss.as.jaxrs"/>
        <extension module="org.jboss.as.idr"/>
        <extension module="org.jboss.as.imx"/>
        <extension module="org.jboss.as.ipa"/>
        <extension module="org.jboss.as.ls"/>
        <extension module="org.jboss.as.logging"/>
        <extension module="org.jboss.as.mail"/>
        <extension module="org.jboss.as.naming"/>
        <extension module="org.jboss.as.pojo"/>
        <extension module="org.jboss.as.remoting"/>
        <extension module="org.jboss.as.sal"/>
        <extension module="org.jboss.as.security"/>
        <extension module="org.jboss.as.transactions"/>
        <extension module="org.jboss.as.weld-v2-features"/>
        <extension module="org.jboss.as.weld"/>
        <extension module="org.wildfly.extension.batch"/>
        <extension module="org.wildfly.extension.bean-validation"/>
        <extension module="org.wildfly.extension.io"/>
        <extension module="org.wildfly.extension.request-controller"/>
        <extension module="org.wildfly.extension.security-manager"/>
        <extension module="org.wildfly.extension.undertow"/>
    </extensions>
    <management>
        <security-realms>
            <security-realm name="ManagementRealm">
                <authentication>
                    <local default-user="$local" skip-group-loading="true"/>
                    <properties path="mgmt-users.properties" relative-to="jboss.server.config.dir"/>
                </authentication>
                <authorization map-groups-to-roles="false">
                    <properties path="mgmt-groups.properties" relative-to="jboss.server.config.dir"/>
                </authorization>
            </security-realm>
            <security-realm name="ApplicationRealm">
                <authentication>
                    <local default-user="$local" allowed-users="*" skip-group-loading="true"/>
                    <properties path="application-users.properties" relative-to="jboss.server.config.dir"/>
                </authentication>
                <authorization>
                    <properties path="application-roles.properties" relative-to="jboss.server.config.dir"/>
                </authorization>
            </security-realm>
        </security-realms>
        <audit-log>
            <formatters>
                <json-formatter name="json-formatter"/>
            </formatters>
            <handlers>
                <file-handler name="file" formatter="json-formatter" relative-to="jboss.server.data.dir" path="audit-log.log"/>
            </handlers>
        </audit-log>
        <logger log-boot="true" log-read-only="false" enabled="false">
    </management>

```

Look for the <Drivers> tag element within the <Datasources> tag element,

```

<?xml version="1.0" ?>
<server xmlns="urn:jboss:domain:3.0">
    <logger>
        <root-logger>
            <level name="INFO"/>
            <handlers>
                <handler name="CONSOLE"/>
                <handler name="FILE"/>
            </handlers>
        </root-logger>
        <formatter name="PATTERN">
            <pattern-formatter pattern="%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n"/>
        </formatter>
        <formatter name="COLOR-PATTERN">
            <pattern-formatter pattern="%K{level}%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n"/>
        </formatter>
    </subsystem>
    <subsystem xmlns="urn:jboss:domain:batch:1.0">
        <job-repository>
            <repository/>
        </job-repository>
        <thread-pool>
            <max-threads count="10"/>
            <keepalive-time time="30" unit="seconds"/>
        </thread-pool>
    </subsystem>
    <subsystem xmlns="urn:jboss:domain:bean-validation:1.0"/>
    <subsystem xmlns="urn:jboss:domain:datasources:3.0">
        <datasources>
            <datasource indi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">
                <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
                <driver>h2</driver>
                <security>
                    <user-name>sa</user-name>
                    <password>sa</password>
                </security>
            </datasource>
            <drivers>
                <driver name="h2" module="com.h2database.h2">
                    <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
                </driver>
            </drivers>
        </datasources>
    </subsystem>
    <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
        <deployment-scanner paths="deployments" relative-to="jboss.server.base.dir" scan-interval="5000" runtime-failure-causes-rollback="$iboss.deployment.scanner.rollback.on.failure=false"/>
    </subsystem>
    <subsystem xmlns="urn:jboss:domain:ee:3.0">
        <spec-descriptor-property-replacement=false></spec-descriptor-property-replacement>
        <concurrent>
            <context-services>
                <context-service name="default" indi-name="java:jboss/ee/concurrency/context/default" use-transaction-setup-provider="true"/>
            </context-services>
            <managed-thread-factories>
                <managed-thread-factory name="default" indi-name="java:jboss/ee/concurrency/factory/default" context-service="default"/>
            </managed-thread-factories>
            <managed-executor-services>
                <managed-executor-service name="default" indi-name="java:jboss/ee/concurrency/executor/default" context-service="default" hung-task-threshold="60000" core-threads="5" max-threads="25" keepalive-time="5000"/>
            </managed-executor-services>
            <managed-scheduled-executor-services>

```

We need to add the PostgreSQL driver here with the XML snippet below

```

150
151 <subsystem xmlns="urn:jboss:domain:bean-validation:1.0">
152   <subsystem xmlns="urn:jboss:domain:core-management:1.0">
153     <subsystem xmlns="urn:jboss:domain:datasources:6.0">
154       <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.datasources.statistics.enabled}">
155         <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
156         <driver>h2</driver>
157         <security>
158           <user-name>sa</user-name>
159           <password>sa</password>
160         </security>
161       </datasource>
162       <drivers>
163         <driver name="h2" module="com.h2database.h2">
164           <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
165         </driver>
166       </drivers>
167     </datasources>
168   </subsystem>
169   <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
170     <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000" runtime-failure-causes-rollback="${jboss.deployment.scanner.rollback.onfailure:false}">
171   </subsystem>
172   <subsystem xmlns="urn:jboss:domain:discovery:1.0">
173   <subsystem xmlns="urn:jboss:domain:distributable-web:2.0" default-session-management="default" default-single-sign-on-management="default">
174     <infinispan-session-management name="default" cache-container="web" granularity="SESSION">
175       <local-affinity/>

```

## add to standalone.xml (under datasources)

```

<drivers>
  <driver name="postgresql" module="org.postgresql">
    <datasource-class>org.postgresql.Driver</datasource-class>
  </driver>
</drivers>

```

```

<logger>
<root-logger>
  <level name="INFO"/>
  <handlers>
    <handler name="CONSOLE"/>
    <handler name="FILE"/>
  </handlers>
</root-logger>
<formatter name="PATTERN">
  <pattern-formatter pattern="%d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c] (%t) %s%n"/>
</formatter>
<formatter name="COLOR-PATTERN">
  <pattern-formatter pattern="%K{level}%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%n"/>
</formatter>
</subsystem>
<subsystem xmlns="urn:jboss:domain:batch:1.0">
<job-repository>
  <in-memory/>
</job-repository>
<thread-pool>
  <max-threads count="10"/>
  <keepalive-time time="30" unit="seconds"/>
</thread-pool>
</subsystem>
<subsystem xmlns="urn:jboss:domain:bean-validation:1.0"/>
<subsystem xmlns="urn:jboss:domain:datasources:3.0">
  <datasources>
    <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">
      <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
      <driver>h2</driver>
      <security>
        <user-name>sa</user-name>
        <password>sa</password>
      </security>
    </datasource>
    <drivers>
      <driver name="h2" module="com.h2database.h2">
        <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
      </driver>
      <driver name="postgresql" module="org.postgresql">
        <datasource-class>org.postgresql.Driver</datasource-class>
      </driver>
    </drivers>
  </datasources>
</subsystem>
<subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
  <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000" runtime-failure-causes-rollback="${jboss.deployment.scanner.rollback.onfailure:false}">
</subsystem>
<subsystem xmlns="urn:jboss:domain:ee:3.0">
  <spec-descriptor-property-replacement>false</spec-descriptor-property-replacement>
  <concurrent>
    <context-services>
      <context-service name="default" jndi-name="java:jboss/ee/concurrency/context/default" use-transaction-setup-provider="true"/>
    </context-services>
    <managed-thread-factories>
      <managed-thread-factory name="default" jndi-name="java:jboss/ee/concurrency/factory/default" context-service="default"/>
    </managed-thread-factories>
    <managed-executor-services>
      <managed-executor-service name="default" jndi-name="java:jboss/ee/concurrency/executor/default" context-service="default" hung-task-threshold="60000"/>
    </managed-executor-services>
  </concurrent>
</subsystem>

```

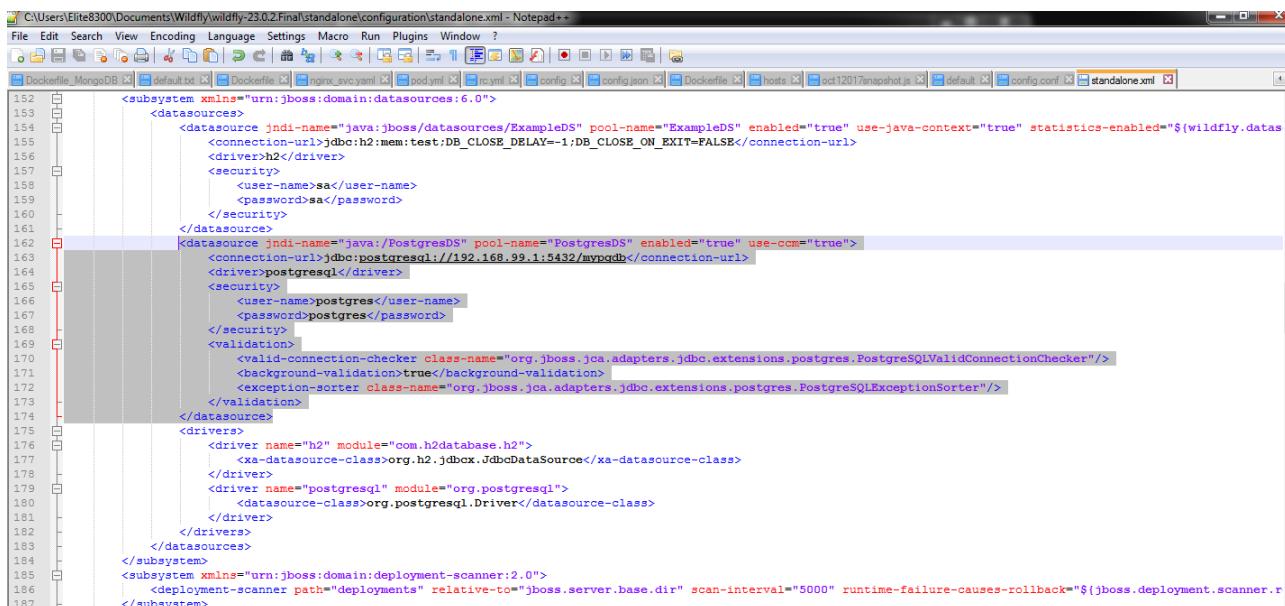
Next, we need to configure the datasource by adding the XML snippet below just above the <drivers> tag section

## add to standalone.xml (under datasources)

```
<datasource jndi-name="java:/PostgresDS" pool-name="PostgresDS" enabled="true" use-ccm="true">
    <connection-url>jdbc:postgresql://192.168.99.100:32770/demo1</connection-url>
        <driver>postgresql</driver>
        <security>
            <user-name>postgres</user-name>
            <password>*****</password>
        </security>
        <validation>
            <valid-connection-checker class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"/>
                <background-validation>true</background-validation>
                <exception-sorter class-
name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"/>
            </validation>
        </datasource>

        </thread-pool>
    </subsystem>
<subsystem xmlns="urn:jboss:domain:bean-validation:1.0"/>
<subsystem xmlns="urn:jboss:domain:datasources:3.0">
    <datasources>
        <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true">
            <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
            <driver>h2</driver>
            <security>
                <user-name>sa</user-name>
                <password>sa</password>
            </security>
        </datasource>
        <datasource jndi-name="java:/PostgresDS" pool-name="PostgresDS" enabled="true" use-ccm="true">
            <connection-url>jdbc:postgresql://192.168.99.100:32770/demo1</connection-url>
            <driver>postgresql</driver>
            <security>
                <user-name>postgres</user-name>
                <password>*****</password>
            </security>
            <validation>
                <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"/>
                <background-validation>true</background-validation>
                <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"/>
            </validation>
        </datasource>
        <drivers>
            <driver name="h2" module="com.h2database.h2">
                <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
            </driver>
            <driver name="postgresql" module="org.postgresql">
                <datasource-class>org.postgresql.Driver</datasource-class>
            </driver>
        </drivers>
    </datasources>
</subsystem>
<subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
```

Restart your Wildfly server to have the new configuration used.



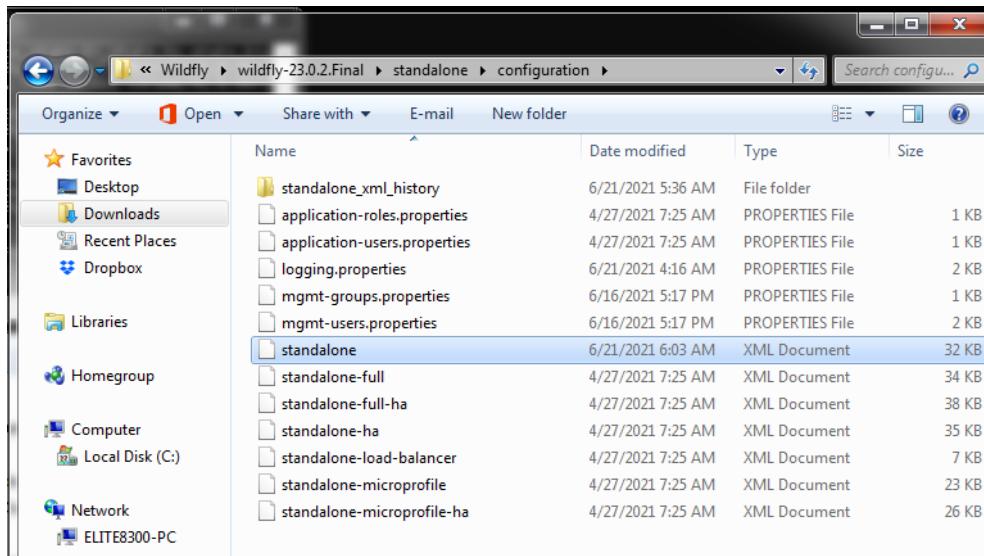
```
152 <subsystem xmlns="urn:jboss:domain:datasources:6.0">
153     <datasources>
154         <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.datas
155             <connection-url>jdbc:h2:mem:test;DB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
156             <driver>h2</driver>
157             <security>
158                 <user-name>sa</user-name>
159                 <password>sa</password>
160             </security>
161         </datasource>
162         <datasource jndi-name="java:/PostgresDS" pool-name="PostgresDS" enabled="true" use-ccm="true">
163             <connection-url>jdbc:postgresql://192.168.99.1:5432/avpgdb</connection-url>
164             <driver>postgresql</driver>
165             <security>
166                 <user-name>postgres</user-name>
167                 <password>postgres</password>
168             </security>
169             <validation>
170                 <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"/>
171                 <background-validation>true</background-validation>
172                 <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"/>
173             </validation>
174         </datasource>
175         <drivers>
176             <driver name="h2" module="com.h2database.h2">
177                 <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
178             </driver>
179             <driver name="postgresql" module="org.postgresql">
180                 <datasource-class>org.postgresql.Driver</datasource-class>
181             </driver>
182         </drivers>
183     </datasources>
184 </subsystem>
185 <subsystem xmlns="urn:jboss:domain:deployment-scanner:2.0">
186     <deployment-scanner path="deployments" relative-to="jboss.server.base.dir" scan-interval="5000" runtime-failure-causes-rollback="${jboss.deployment.scanner.r
187 </subsystem>
```

You need to use the correct URI for your running PostgreSQL instance in the above.

```

<subsystem xmlns="urn:jboss:domain:datasources:6.0">
    <datasources>
        <datasource jndi-name="java:jboss/datasources/ExampleDS" pool-name="ExampleDS" enabled="true" use-java-context="true" statistics-enabled="${wildfly.datasources.statistics-enabled:${wildfly.statistics-enabled:false}}">
            <connection-url>jdbc:h2:mem:testDB_CLOSE_DELAY=-1;DB_CLOSE_ON_EXIT=FALSE</connection-url>
            <driver>h2</driver>
            <security>
                <user-name>sa</user-name>
                <password>sa</password>
            </security>
        </datasource>
        <datasource jndi-name="java:PostgresDS" pool-name="PostgresDS" enabled="true" use-cmm="true">
            <connection-url>jdbc:postgresql://192.168.99.1:5432/mypgdb</connection-url>
            <driver>postgresql</driver>
            <security>
                <user-name>postgres</user-name>
                <password>postgres</password>
            </security>
            <validation>
                <valid-connection-checker class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLValidConnectionChecker"/>
                <background-validation>true</background-validation>
                <exception-sorter class-name="org.jboss.jca.adapters.jdbc.extensions.postgres.PostgreSQLExceptionSorter"/>
            </validation>
        </datasource>
    <drivers>
        <driver name="h2" module="com.h2database.h2">
            <xa-datasource-class>org.h2.jdbcx.JdbcDataSource</xa-datasource-class>
        </driver>
        <driver name="postgresql" module="org.postgresql">
            <datasource-class>org.postgresql.Driver</datasource-class>
        </driver>
    </drivers>
    <datasources>
    </subsystem>

```



The edited **standalone.xml** file is within your Wildfly folder as above

The screenshot shows the WildFly 9.0.1.Final management interface. The top navigation bar includes links for Chrome, File, Edit, View, History, Bookmarks, People, Window, Help, and a user icon. The address bar shows the URL [127.0.0.1:9990/console/App.html#/home](http://127.0.0.1:9990/console/App.html#/home). The main content area displays the WildFly logo and a "Find More Resources" sidebar.

**Find More Resources**

- General Resources
  - [WildFly Home](#)
  - [WildFly Documentation](#)
  - [Admin Guide](#)
  - [Model Reference Documentation](#)
  - [Browse Issues](#)

**WildFly**

**View and Manage Settings**

- Deployments**: Add and manage deployments.
- Runtime**: Monitor server status, retrieve diagnostic information and perform other operational tasks.
- Configuration**: Configure your server profiles, including the attributes and settings that define subsystems and resources available to your servers.
- Access Control**: Perform administrative tasks for your server, including role-based access control.

JDBC DataSources

JDBC datasource configurations.

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
PostgresDS	java:/PostgresDS	<input checked="" type="checkbox"/>

Attributes Connection Pool Security Properties Validation Timeouts Statements

**PostgresDS**

Edit

Connection URL: jdbc:postgresql://192.168.99.100:32770/demo1

New Connection Sql:

Transaction Isolation:

Use JTA?: true

Use CCM?: true

Test Connection Need Help?

Verify your connections in the management console

JDBC DataSources

JDBC datasource configurations.

Name	JNDI	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	<input checked="" type="checkbox"/>
PostgresDS	java:/PostgresDS	<input checked="" type="checkbox"/>

Attributes Connection Pool

**PostgresDS**

Edit

Connection URL: jdbc:postgre...

Test Connection Need Help?

OK



Let us now see how to add the PostgreSQL datasource to Wildfly via the CLI

```
bin - bash - 80x24
Last login: Thu Jul 30 20:58:07 on ttys000
Daniels-MacBook-Pro:bin dspiess$
```

- 1) CLI commands can be scripted
- 2) Script can run while server is running

### CLI commands

1 ./jboss-cli.sh --connect controller=127.0.0.1

Start the Wildfly application server using the above command within the Wildfly folder

```
bin - java - 80x24
Last login: Thu Jul 30 20:58:07 on ttys000
Daniels-MacBook-Pro:bin dspiess$ ./jboss-cli.sh --connect controller=127.0.0.1
```

Then connect to the running Wildfly application server and install the JDBC JAR file

### CLI commands

1 ./jboss-cli.sh --connect controller=127.0.0.1  
2 module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.api

This loads the JAR file by adding a module

```
bin - java - 80x24
Last login: Thu Jul 30 20:58:07 on ttys000
Daniels-MacBook-Pro:bin dspiess$ ./jboss-cli.sh --connect controller=127.0.0.1
[standalone@127.0.0.1:9990 /] module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.ap
i
[standalone@127.0.0.1:9990 /]
```

### CLI commands

1 ./jboss-cli.sh --connect controller=127.0.0.1  
2 module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.api  
3 /subsystem=datasources/jdbc-driver=postgres:add(driver-name="postgres",driver-module-name="org.postgres",driver-class-name=org.postgresql.Driver)

Then we add the driver to the datasource subfolder using the command above

```

Last login: Thu Jul 30 20:58:07 on ttys000
Daniels-MacBook-Pro:bin dspiess$ ./jboss-cli.sh --connect controller=127.0.0.1
[standalone@127.0.0.1:9990 /] module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.api
[standalone@127.0.0.1:9990 /] /subsystem=datasources/jdbc-driver=postgres:add(driver-name="postgres",driver-module-name="org.postgres",driver-class-name=org.postgresql.Driver)
{"outcome" => "success"}
[standalone@127.0.0.1:9990 /]

```

## CLI commands

- 1 . ./jboss-cli.sh --connect controller=127.0.0.1
- 2 module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.api
- 3 /subsystem=datasources/jdbc-driver=postgres:add(driver-name="postgres",driver-module-name="org.postgres",driver-class-name=org.postgresql.Driver)
- 4 data-source add --jndi-name=java:/PostgresDS --name=PostgresPool --connection-url=jdbc:postgresql://192.168.99.100:32770/demo1 --driver-name=postgres --user-name=postgres --password=

Finally, we create the datasource that uses the JDBC driver we just installed

```

Last login: Thu Jul 30 20:58:07 on ttys000
Daniels-MacBook-Pro:bin dspiess$ ./jboss-cli.sh --connect controller=127.0.0.1
[standalone@127.0.0.1:9990 /] module add --name=org.postgres --resources=~/Downloads/postgresql-9.4-1201.jdbc4.jar --dependencies=javax.api,javax.transaction.api
[standalone@127.0.0.1:9990 /] /subsystem=datasources/jdbc-driver=postgres:add(driver-name="postgres",driver-module-name="org.postgres",driver-class-name=org.postgresql.Driver)
{"outcome" => "success"}
[standalone@127.0.0.1:9990 /] data-source add --jndi-name=java:/PostgresDS --name=PostgresPool --connection-url=jdbc:postgresql://192.168.99.100:32770/demo1 --driver-name=postgres --user-name=postgres --password=
[standalone@127.0.0.1:9990 /]

```

The screenshot shows the WildFly 9.0.1.Final management interface. The top navigation bar includes links for Analytics, \$Scope, \$Watch..., Design Patterns, Object Oriented..., Pattern Overview, CLI Recipes - W..., Configuring a da..., How to add Pos..., WildFly Full Man..., Add new Datas..., and a search bar. The main content area has tabs for Home, Deployments, Configuration, Runtime, Access Control, and Patching. The Home tab is selected. It displays sections for WildFly, View and Manage Settings, Deployments, Runtime, Common Tasks, and a sidebar with Find More Resources and Get Help sections.

WildFly 9.0.1.Final

Home Deployments Configuration Runtime Access Control Patching

**Configuration**

**Subsystem**

Subsystems > jca

Interfaces > Datasources

Socket Binding > Resource Adapters

Paths > Mail

System Properties > Transactions

EJB 3

**Datasources**

The Datasource subsystem allows you to create and configure datasources and manage JDBC database drivers.

WildFly 9.0.1.Final

Configuration: Subsystems > Subsystem: Datasources

DATASOURCES XA DATASOURCES

JDBC Datasources

JDBC datasource configurations.

Name	Type	Enabled?
ExampleDS	java:jboss/datasources/ExampleDS	✓
PostgresPool	java:jboss/datasources/PostgresPool	✓

Add Remove Disable

Test Connection

Successfully created JDBC connection.  
Successfully connected to database PostgresPool.

OK Need Help?

This works also and can be implemented using a script to be run

## Management Console

Requires Access to Management Console



## standalone.xml

## CLI

Configuration can be source controlled



Requires Reboot



Configuration changes at runtime

