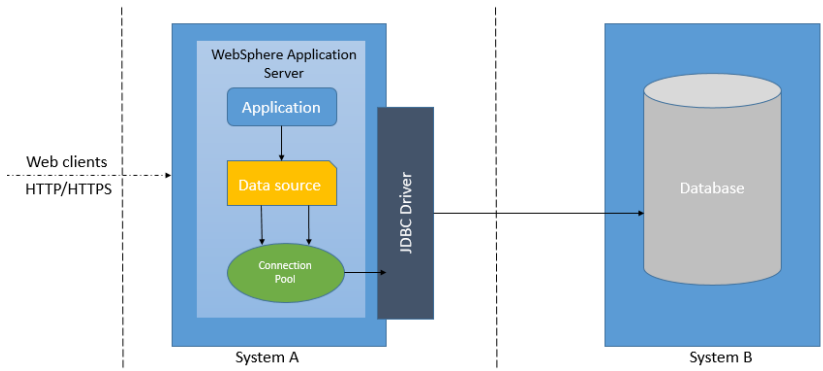
CHAPTER 7: CONNECT TO A DATABASE

Theory

Business applications running on WebSphere Application Server requires access to database systems. In order to access to databases, we need to define a data source for each database.

For better understanding of the tasks required to access databases, we need to understand following terms:

- **JDBC**, Java Database Connectivity, is a Java-based API technology to access databases. You can connect to a database, query and change data in a database. There are 2 types of JDBC drivers in WebSphere Application Server, version 2 driver (requires database client to connect to the database server) and version 4 driver (can directly connect to the database).
- **Data source**, is referred to the name of the configuration properties of the database in order to connect and run queries.



- **Connection pool**, is a configuration object that provides a set of connections to databases for the applications. When an application requires access to a database, it will use an existing connection from the pool and connection pool will create a new connection if there is no pooled connections available. You can set minimum and maximum number of connections for the pool to prevent overhead related with database connection requests.

- **JDBC Provider,** supplies the specific JDBC driver class to a specific database vendor. To create a data source, we need to associate a data source with the JDBC provider.
- **JNDI,** Java Naming Directory Interface, is a Java API that gives applications access to database connections.
- **J2C authentication alias,** is a feature that encrypts the password used by the adapter to access to a database.

In order to provide access to a database from an application that runs on WebSphere Application Server, you need to follow 2 basic steps:

0. Create and J2C authentication alias to store and encrypt credentials which will be used to connect to the database.
1. Create a JDBC provider that contains information of database drivers, type of access and location of the files needed for the implementation.
2. Create a data source that defines which JDBC driver to use, database name and location, and other connection properties.

AIM

In this lab exercise, you will enable access to applications from the WebSphere Application Server. In order to complete the exercise, you need to have following information beforehand:

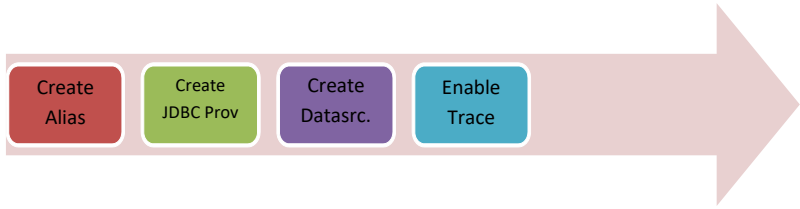
- A running database instance
- Hostname or IP address of the server where the database runs
- Port number to connect to the database
- Sample database name
- Username and password to connect to the database server and the database.

Note: If database is not available, still lab guide can be completed. Although, you won't be able to test connection with database at the end of this lab.

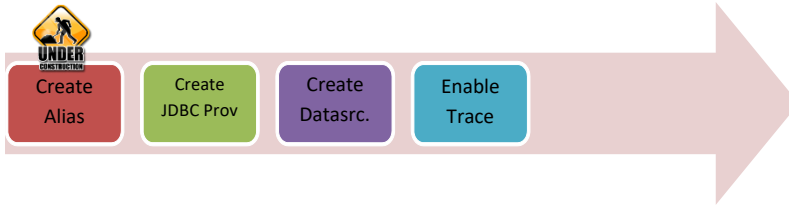
You need to follow the tasks below:

- Task 1: Create an authentication alias
- Task 2: Create JDBC provider
- Task 3: Create data source
- Task 4: Enable JDBC trace logs

Lab Exercise 7: CONNECT TO A DATABASE

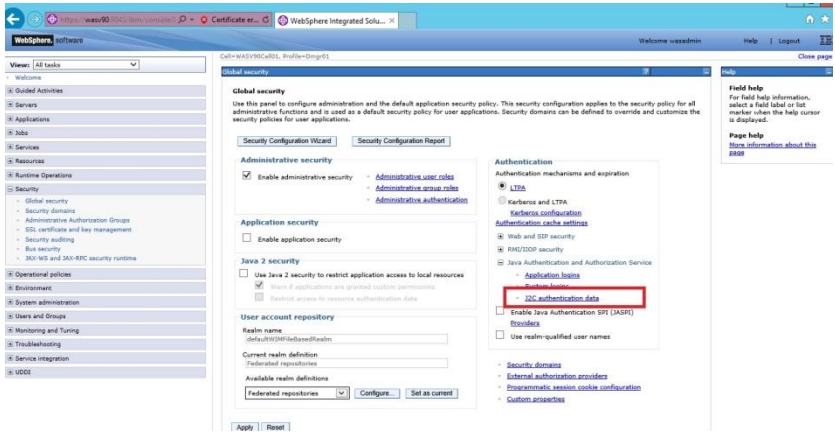


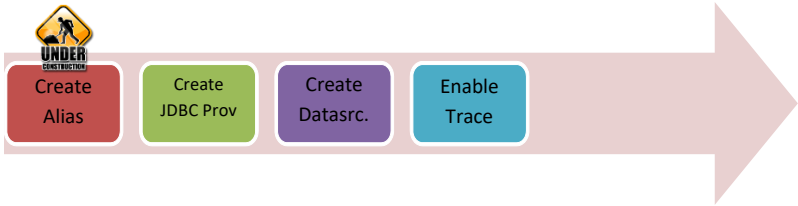
1. **Create an authentication alias**
2. **Create JDBC provider**
3. **Create data source**
4. **Enable JDBC trace logs**



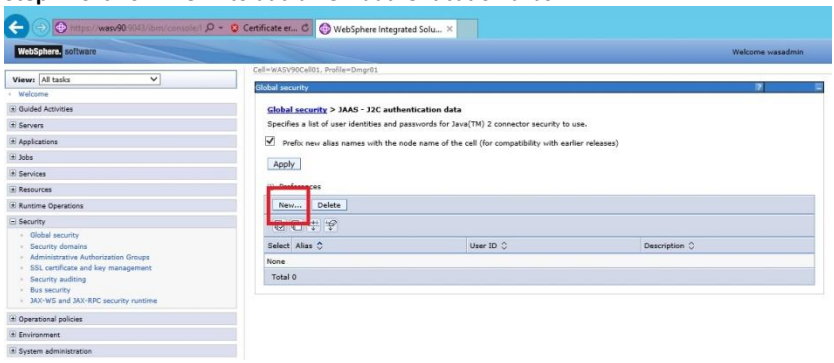
Task 1: Create an authentication alias

Step 1: Login to admin console and navigate to “Security>Global security”. Under the “Authentication” part, located on the right, expand “Java Authentication and Authorization” menu and click on “J2C authentication data”.

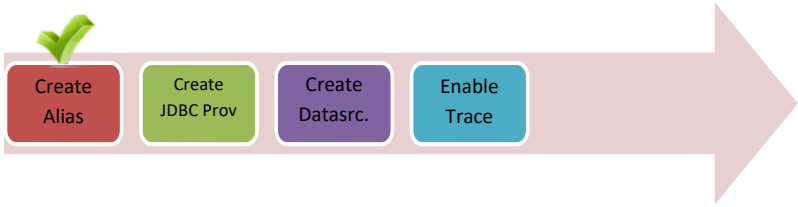




Step 2: Click on “New” to add a new authentication alias.

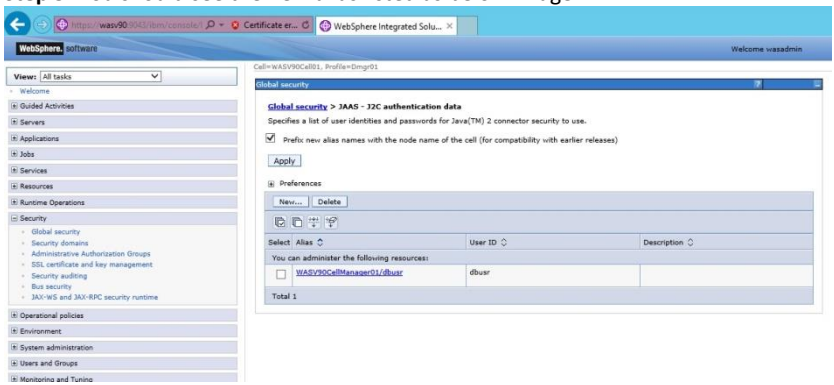


Step 3: Give an alias name and enter the credentials for database connection, then click “OK”.

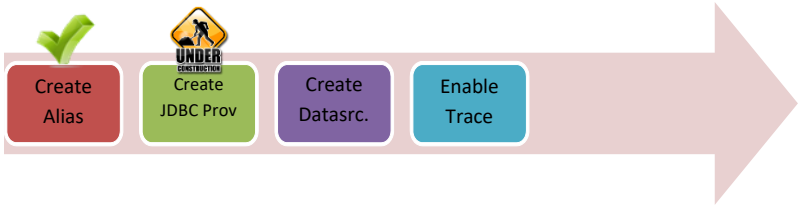


Step 4: Click “Save” to write changes directly to the master configuration.

Step 5: You should see the new alias listed as below image.

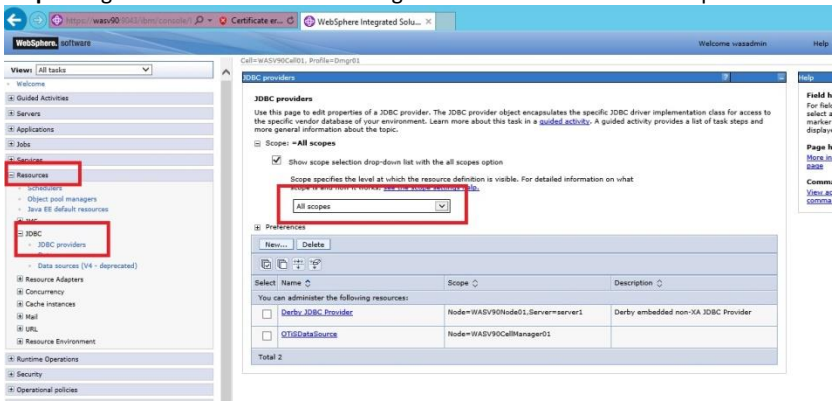


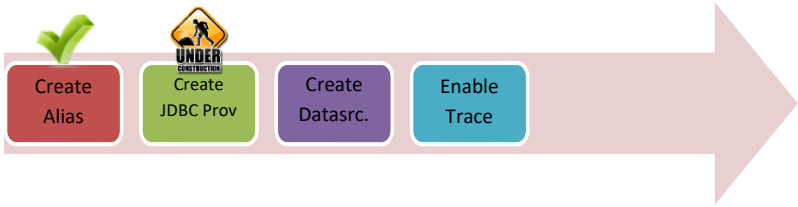
Task 1 is complete!



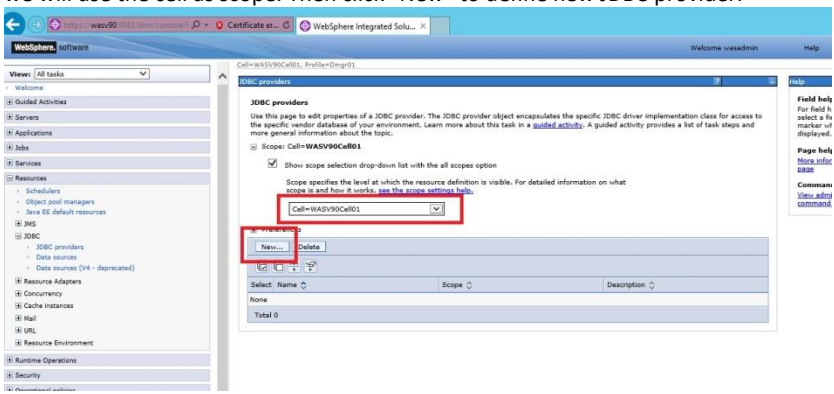
Task 2: Create JDBC provider

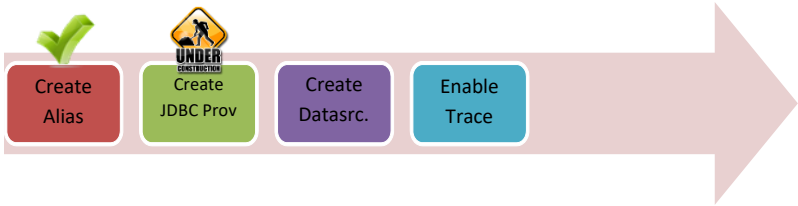
Step 1: Login to admin console and navigate to “Resources>JDBC>JDBC provides”.





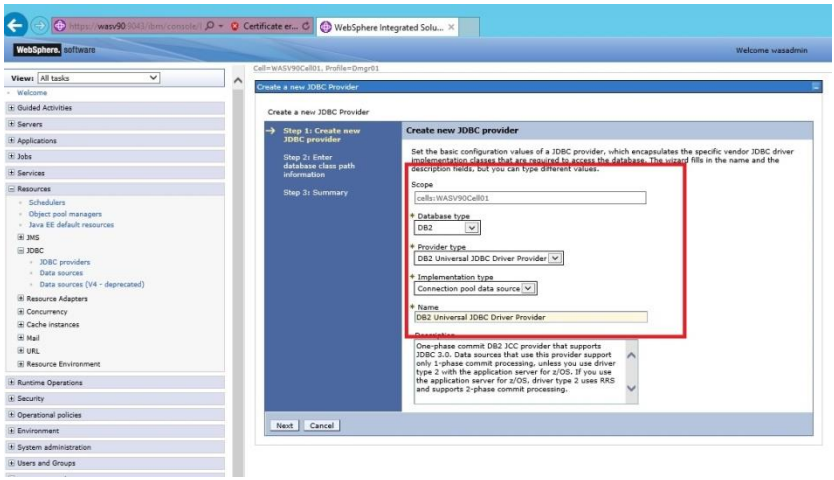
Step 2: You need to change the scope depending on your needs. For this example, we will use the cell as scope. Then click “New” to define new JDBC provider.

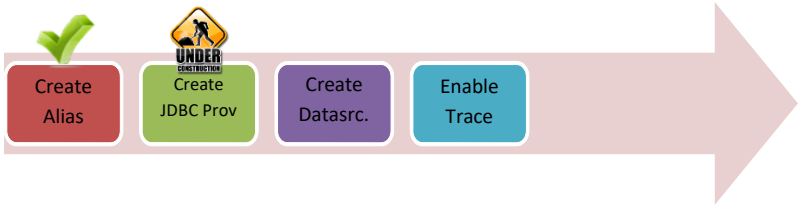




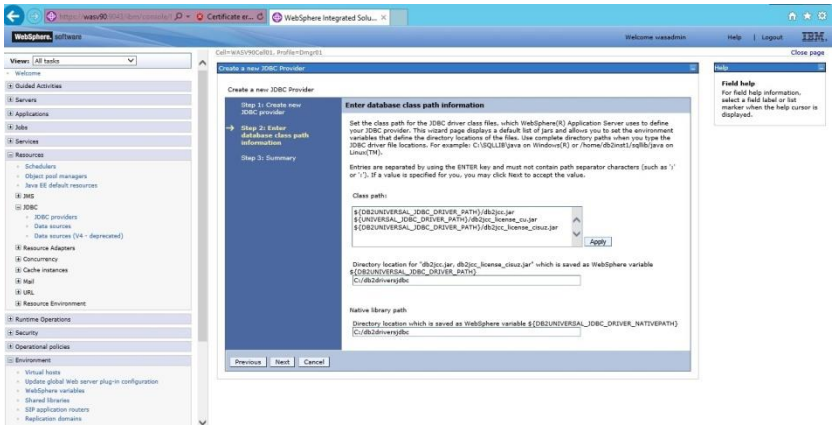
Step 3: In this step you need to configure following properties:
Database type: DB2
Provider type: DB2 Universal JDBC Driver Provider
Implementation type: Connection pool data source

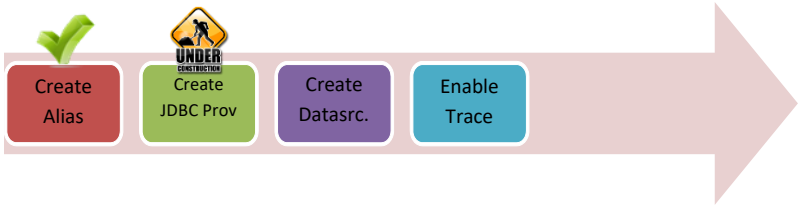
Click “Next” to continue.



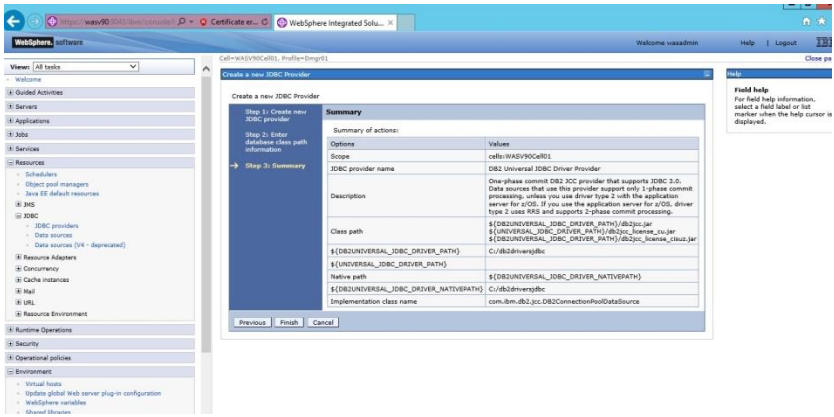


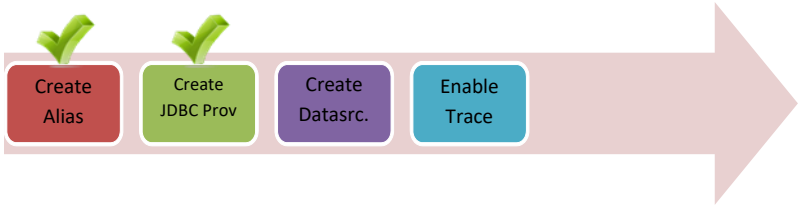
Step 4: You need to copy database drivers to the server where we have the deployment manager installed. As an example, we stored DB2 drivers under **"/headless/Desktop/websphere/jars**". In this step, we need to configure the location of the drivers as shown below and click "Next".



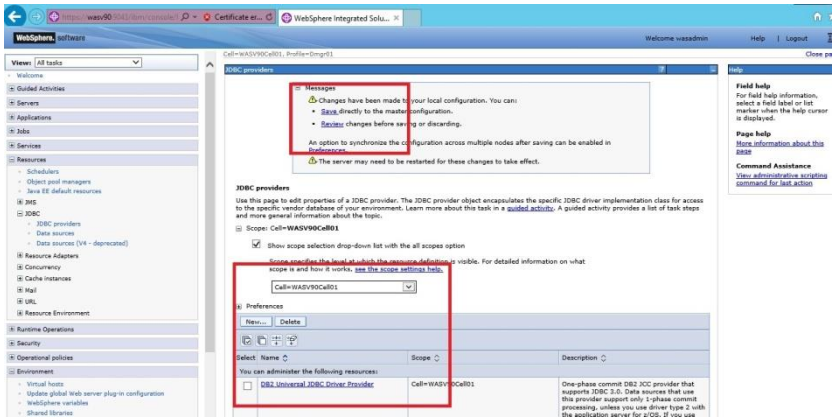


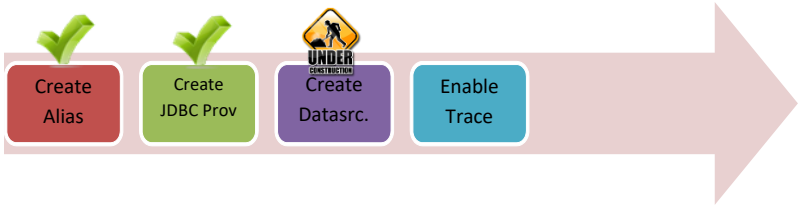
Step 5: Review the summary of options and then click “Finish”.





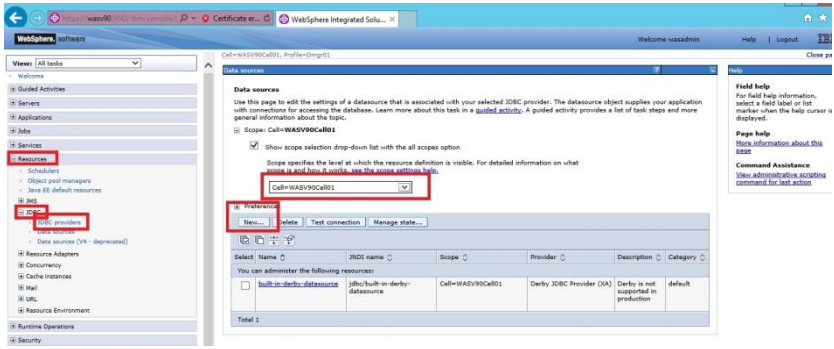
Step 6: Click “Save” to write the changes to master repository. You should see the newly created JDBC provider.

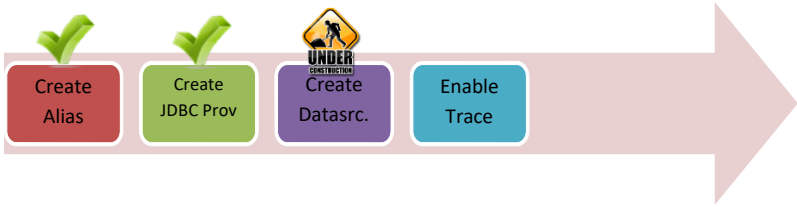




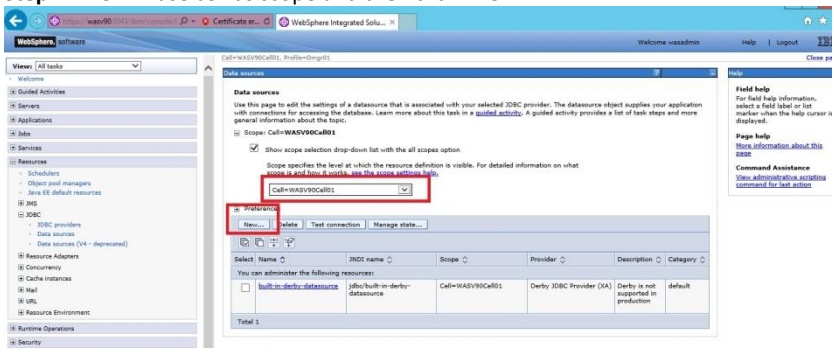
Task 3: Create data source

Step 1: Navigate to “Resources>JDBC>Data sources”. Change the scope according to your needs.

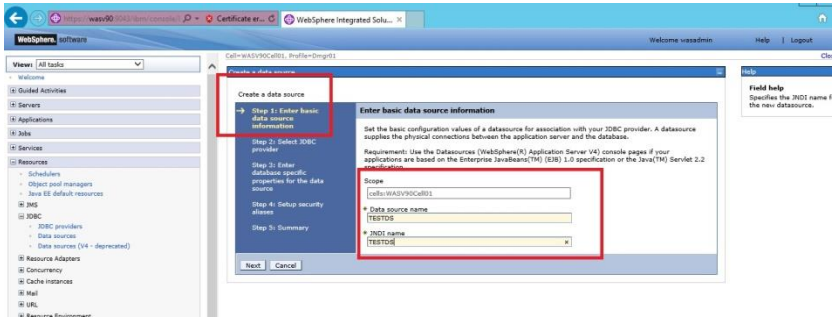


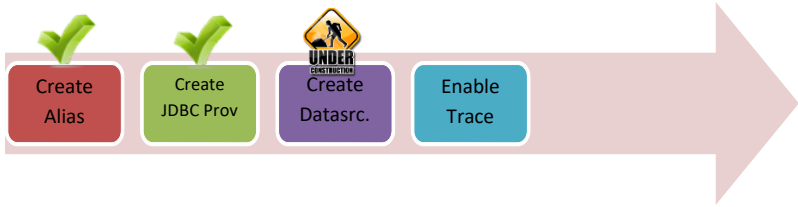


Step 2: We will use cell as scope and then click “New”.

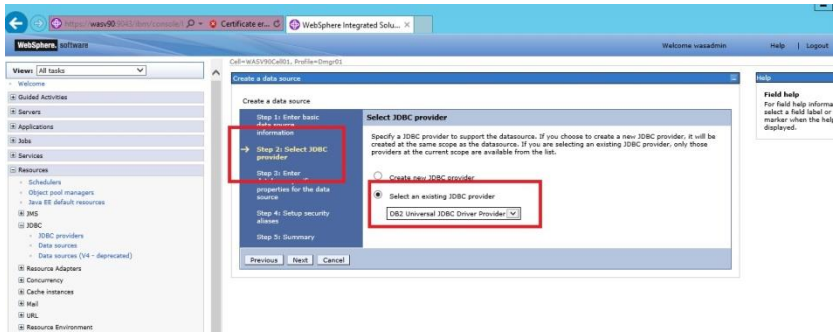


Step 3: Enter the data source (TESTDS) and JNDI (TESTDS) names and then click “Next”.

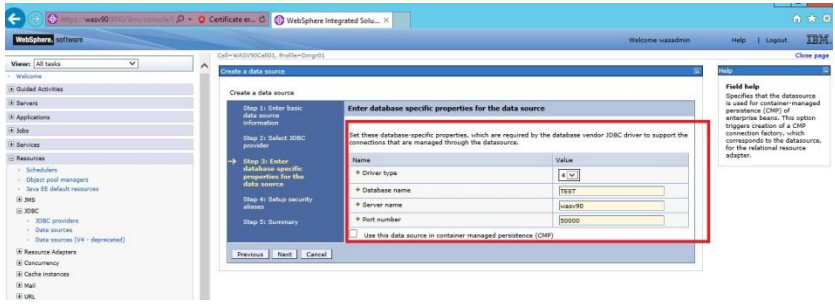


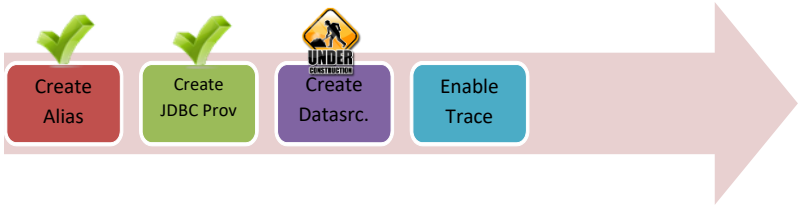


Step 4: Select “Select an existing JDBC Provider” and from the list sselect “DB2 Universal JDBC Driver Provider” then click “Next”.

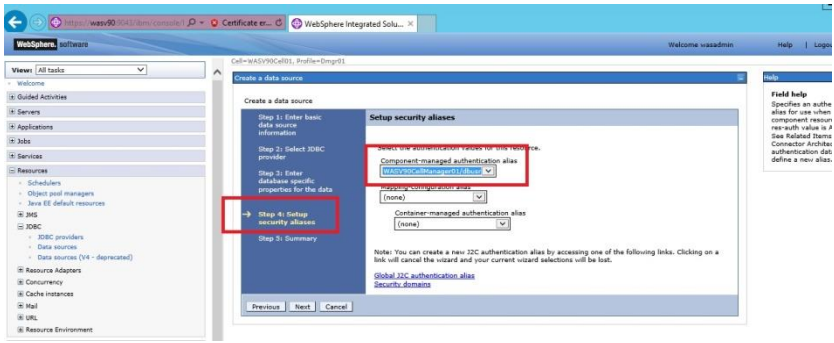


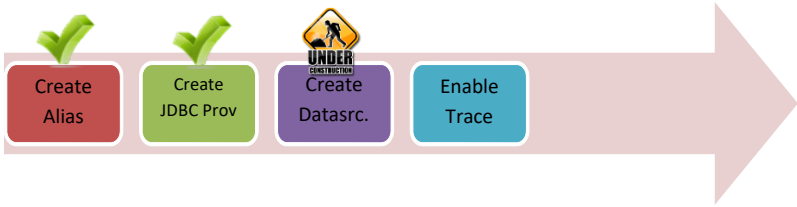
Step 5: You need to enter database properties (driver type should be 4, database name, database host, database port) and click “Next”.



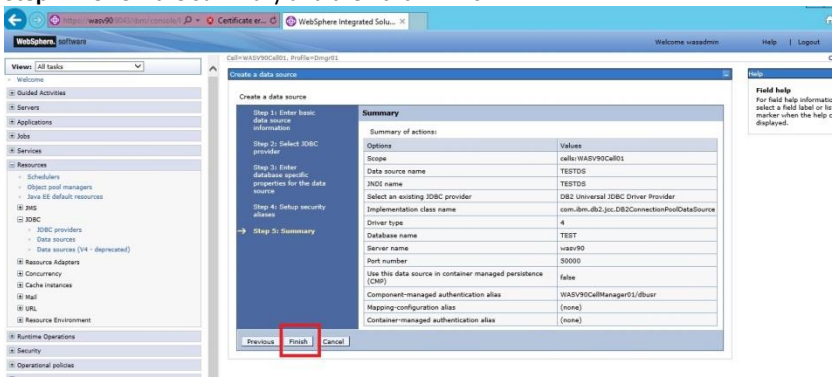


Step 6: Select authentication alias we created in the first task for the “Component-managed authentication alias” and “Container-managed authentication alias”, then click “Next”.

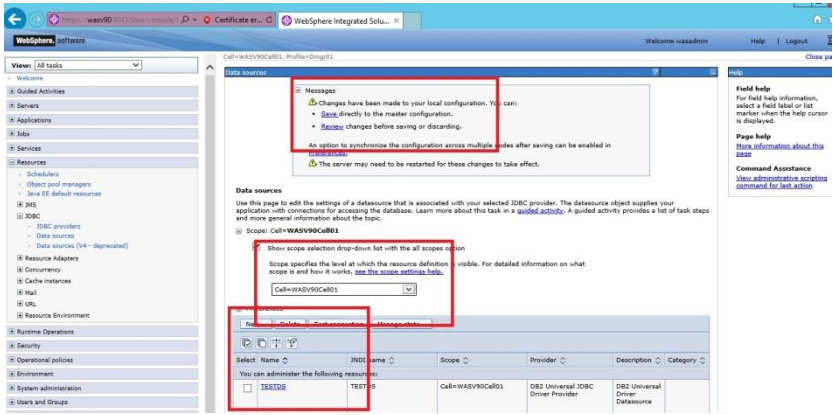


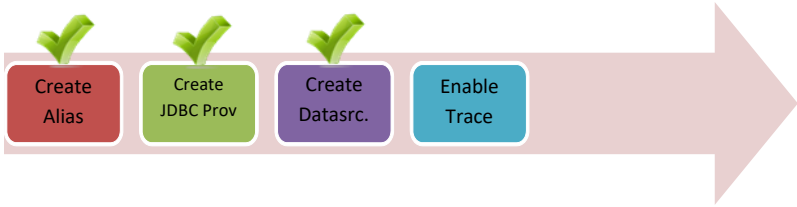


Step 7: Review the summary and then click “Finish”.

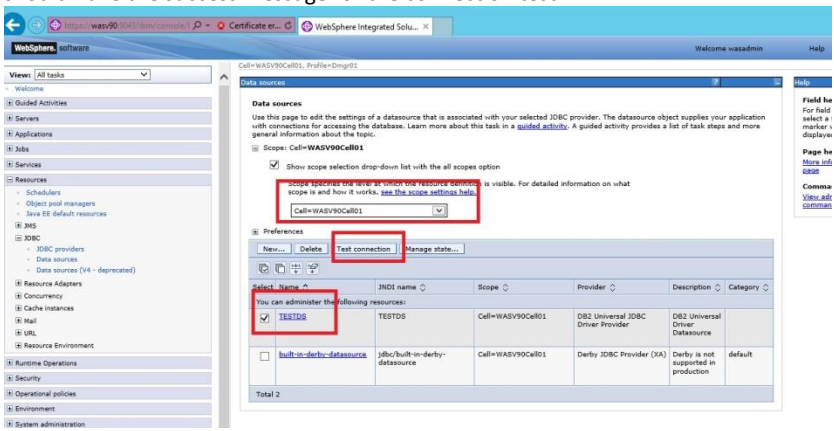


Step 8: Click “Save” to write changes.

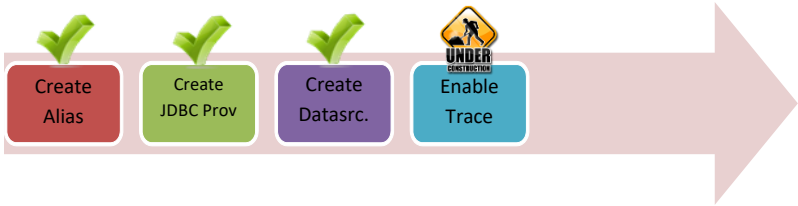




Step 9: Select the data source recently added and click “Test connection”. You should have the success message for the connection test.

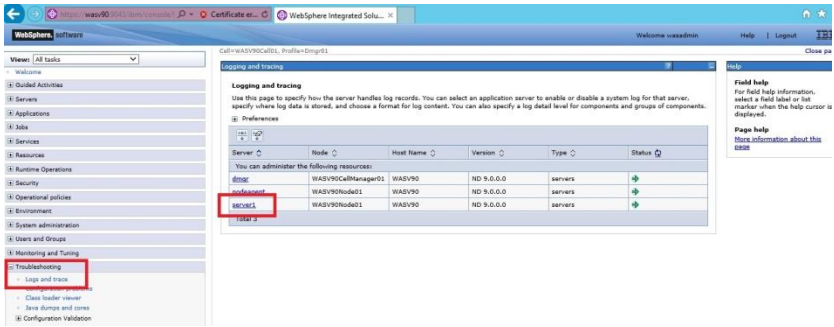


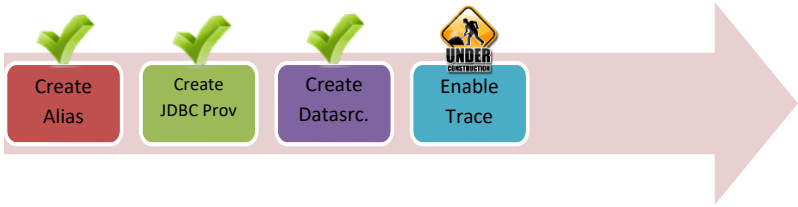
Task 3 is complete!



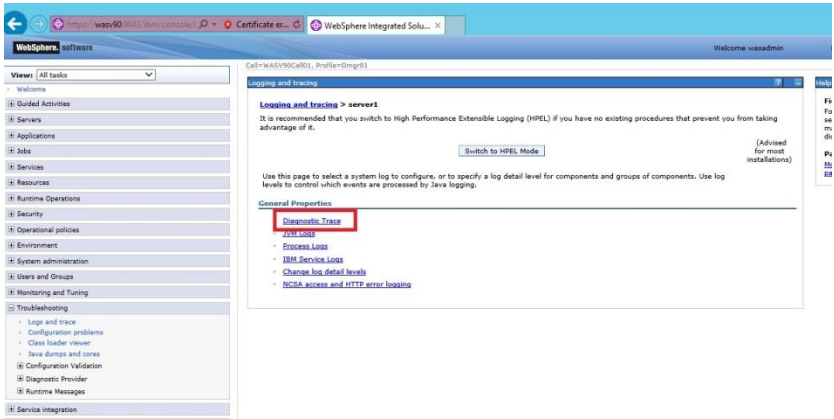
Task 4: Enable JDBC trace logs

Step 1: Navigate to “Troubleshooting>Logs and trace” and then click on “server1”.

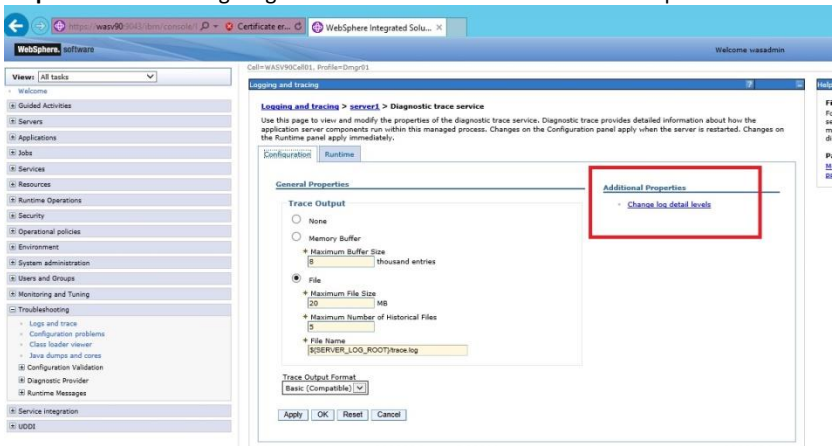


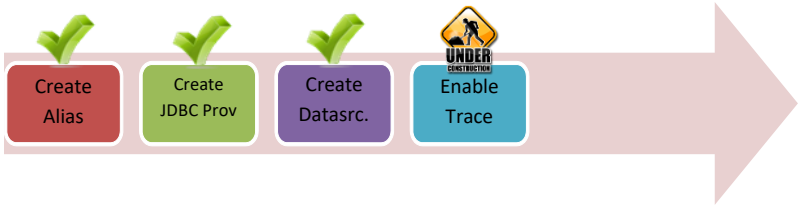


Step 2: Click on “Diagnostic Trace”.



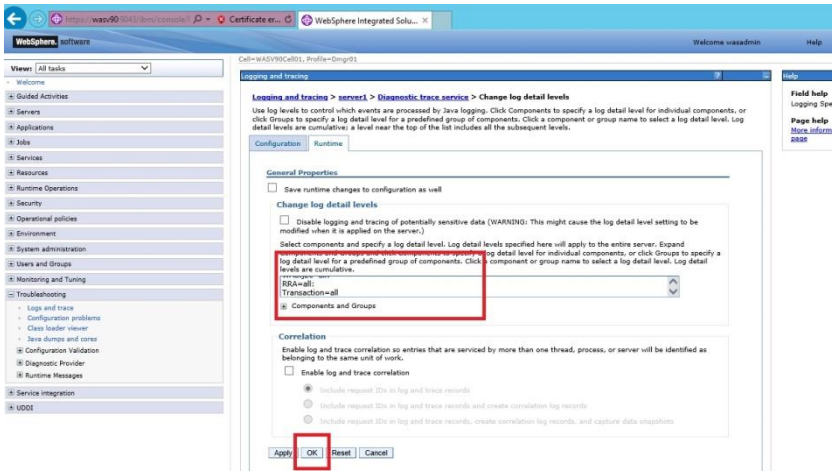
Step 3: Click on “Change log detail levels” under the “Additional Properties”.

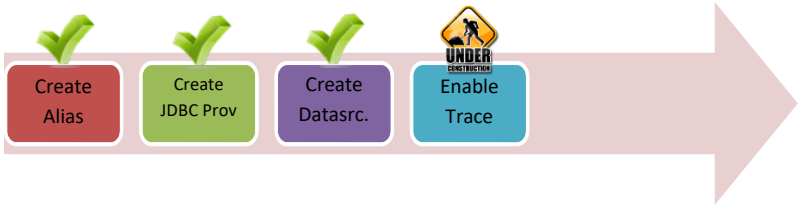




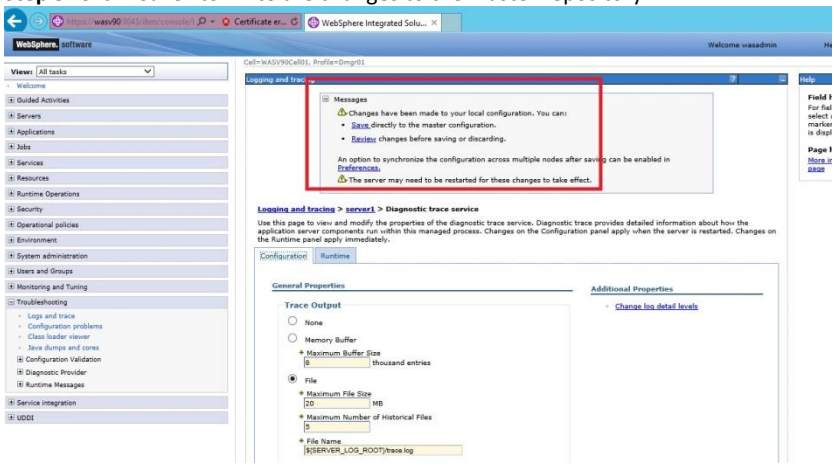
Step 4: In the “Change log detail levels”, put the following into the text box and then click “OK”.

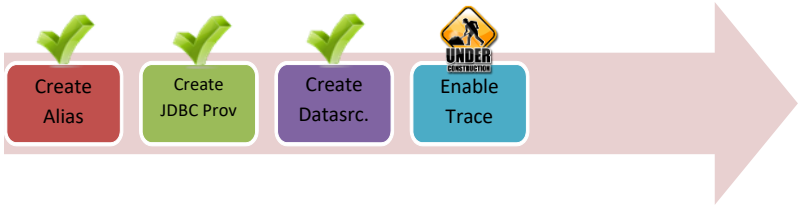
**=info:*
WAS.j2c=all:
RRA=all:
Transaction=all



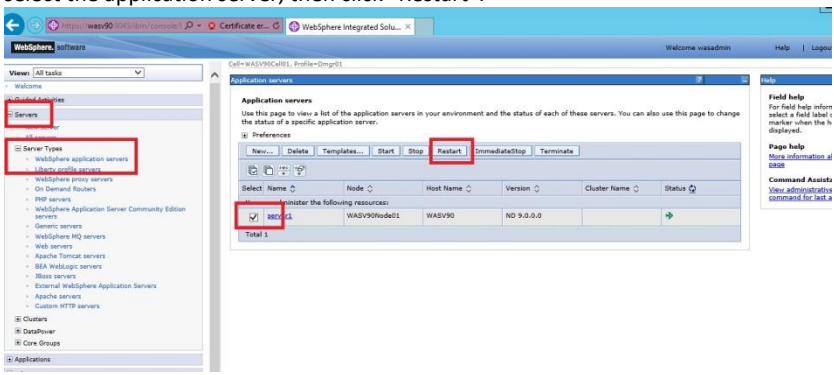


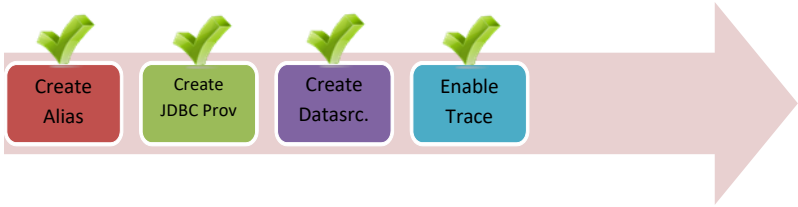
Step 5: Click “Save” to write the changes to the master repository.



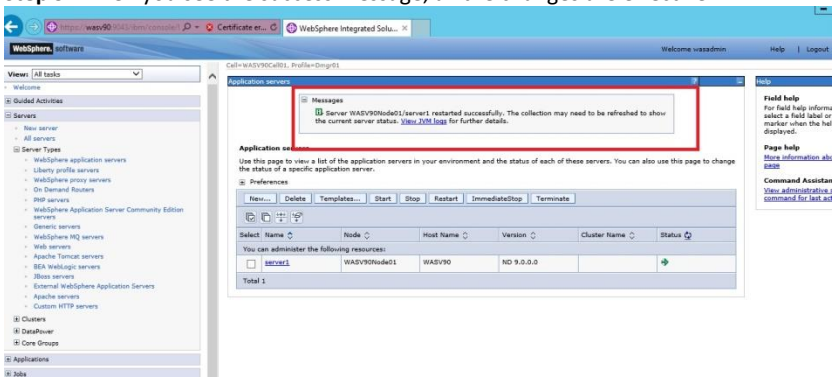


Step 6: Navigate to “Servers>Server Types>WebSphere application servers” and select the application server, then click “Restart”.





Step 9: When you see the success message, all the changes are effective.



Task 4 is complete!

SUMMARY

Business applications running on WebSphere Application Server requires access to database systems. In order to access to databases, we need to define a data source for each database. You need to create a JDBC provider that contains information of database drivers, type of access and location of the files needed for the implementation and to create a data source that defines which JDBC driver to use, database name and location, and other connection properties.

REFERENCES

- http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wbit.sample.tech.2.doc/enterprisedisc/topics/tcrtalias.html
- http://pic.dhe.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=%2Fcom.ibm.websphere.nd.multiplatform.doc%2Finfo%2Fae%2Fae%2Ftdat_tccrtprovds.html
- http://publib.boulder.ibm.com/infocenter/wsdoc400/v6r0/index.jsp?topic=/com.ibm.websphere.iseries.doc/info/ae/ae/rtrb_jdbccomp.html

INDEX