

Install and configure Postgresql-15, PL/Java, and DB client on RHEL 8.x

▪ Install and configure Postgresql-15

1. Update the Server:

```
dnf update -y
```

2. Install the prerequisites:

```
dnf install pg_top_15 gcc-c++  
dnf install java-17-openjdk java-17-openjdk-devel  
dnf install redhat-rpm-config -y  
dnf --enablerepo=ol8_codeready_builder install perl-IPC-Run
```

3. Install the repository RPM:

```
dnf install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-8-x86_64/pgdg-redhat-repo-latest.noarch.rpm
```

4. Disable the built-in PostgreSQL module:

```
dnf -qy module disable postgresql
```

5. Install PostgreSQL:

```
dnf install -y postgresql15 postgresql15-server postgresql15-contrib postgresql15-devel
```

6. Optionally initialize the database and enable automatic start:

```
sudo /usr/pgsql-15/bin/postgresql-15-setup initdb  
sudo systemctl enable postgresql-15  
sudo systemctl start postgresql-15  
sudo systemctl status postgresql-15
```

7. Configure “pg_hba.conf” according to your environment:

***Restart Required to apply the changes**

```
(vim /var/lib/pgsql/15/data/pg_hba.conf)
```

Ex.

```
host all all 192.168.8.0/24 md5
```

8. Configure “postgresql.conf” according to your environment:

***Restart Required to apply the changes**

```
(vim /var/lib/pgsql/15/data/postgresql.conf)
```

| | | |
|----------------------------|-------------------------|--|
| listen_addresses | = '*' | -- allowing connection from all IPs. |
| max_connections | = < appropriate value > | -- allowing max number of connections to a database. |
| shared_buffers | = < appropriate value > | -- (25% of total server memory) |
| work_mem | = < appropriate value > | -- Working memory for sort operation |
| log_min_duration_statement | = < appropriate value > | -- log queries exceeding specified milliseconds. |
| log_line_prefix | = '%m %u ' | -- add time and user information as a prefix in postgres log file. |
| lock_timeout | = <appropriate value > | -- abort any connection waiting to acquire a lock longer than specified time. (milliseconds) |

| Parameters | Small Database | Medium Database | Large Database |
|----------------------------|----------------|-----------------|-----------------|
| listen_addresses | * | * | * |
| max_connections | 500 | 500 | 1000 |
| shared_buffers | 512 (MB) | 1GB | 2GB |
| work_mem | 64MB | 128MB | 256MB |
| log_min_duration_statement | 1000 | 1000 | 1000 |
| log_line_prefix | <%r %m %u > | <%r %m %u > | <%r %m %u > |
| lock_timeout | 180000 (3mins) | 180000 (3mins) | 300000 (5 mins) |

▪ Build PL/Java

1. Update the Server:

```
dnf update -y
```

2. Install Prerequisites:

```
dnf install gcc-c++
dnf install openssl-devel
dnf install redhat-rpm-config
dnf --enablerepo=ol8_codeready_builder install perl-IPC-Run -y
dnf install java-17-openjdk java-17-openjdk-devel
dnf install maven
dnf install git
```

3. Validate with “mvn -version” command:

```
mvn -version
```

4. Disable the built-in PostgreSQL module:

```
dnf -qy module disable postgresql
```

5. Install “Postgresql-devel” package:

```
dnf install postgresql15-devel -y
```

6. check the “pg_config” status:

```
pg_config
```

If command not found, export path of postgres /usr/pgsql-15/bin/

Ex.

```
export PATH=$PATH:/usr/pgsql-15/bin/
```

6. check the java version:

```
java -version
```

- If the java version not found or selected another version, export correct jdk-17 path.

Ex.

```
export JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.8.0.7-2.el8.x86_64/
```

```
export PATH=$JAVA_HOME/bin:$PATH
```

- Change default java version or check java installed path,

```
alternatives --config java
```

7. Download the latest pljava version:

- URL: <https://github.com/tada/pljava/releases>

```
wget https://github.com/tada/pljava/archive/refs/tags/V1\_6\_5.tar.gz
```

8. Unzip the.tar.gz file:

```
tar -xvzf V1_6_5.tar.gz
```

9. Enter to the unzipped folder:

```
cd pljava-1_6_5/
```

10. Start the build using maven:

mvn clean install

```
[INFO] PostgreSQL PL/Java 1.6.5 ..... SUCCESS [ 26.769 s]
[INFO] PL/Java API ..... SUCCESS [ 12.524 s]
[INFO] PL/Java backend Java code ..... SUCCESS [ 4.065 s]
[INFO] PL/Java PGXS ..... SUCCESS [ 14.874 s]
[INFO] PL/Java backend native code ..... SUCCESS [ 11.599 s]
[INFO] PL/Java Ant tasks ..... SUCCESS [ 5.652 s]
[INFO] PL/Java examples ..... SUCCESS [ 3.039 s]
[INFO] PL/Java packaging 1.6.5 ..... SUCCESS [ 2.560 s]
```

11. If it shows errors, compile it with maven debug mode to find errors:

mvn -X clean install

12. Once the compile is success, switch to “/pljava-1_6_5/pljava-packaging/target” directory:

If the compile is succeeded “.jar” file shows like below

```
drwxr-xr-x. 2 root root      28 Sep 14 18:47 antrun
drwxr-xr-x. 3 root root     160 Sep 14 18:47 classes
drwxr-xr-x. 3 root root      25 Sep 14 18:47 generated-sources
drwxr-xr-x. 3 root root      35 Sep 14 18:47 maven-status
-rw-r--r--. 1 root root 2240945 Sep 14 18:47 pljava-pg15.jar
```

13. Now, run Pljava-packaging/pg11xxx jar file so that it will extract/produce needed files in PostgreSQL Library directory:

java -jar pljava-pg15.jar

```
[root@dc-db-01-prod target]# java -jar pljava-pg15.jar
/usr/pgsql-15/lib/libpljava-so-1.6.5.so as bytes
/usr/pgsql-15/share/pljava/pljava-1.6.5.jar as bytes
/usr/pgsql-15/share/pljava/pljava-api-1.6.5.jar as bytes
/usr/pgsql-15/share/pljava/pljava-examples-1.6.5.jar as bytes
/usr/pgsql-15/share/extension/pljava.control as lines (ASCII)
/usr/pgsql-15/share/pljava/pljava--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--unpackaged--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.6.4--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.6.3--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.6.2--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.6.1--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.6.0--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.8--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.7--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.6--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.5--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.4--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.3--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.2--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.1--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.1-BETA3--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.1-BETA2--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.1-BETA1--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.0--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.0-BETA3--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.0-BETA2--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--1.5.0-BETA1--1.6.5.sql as lines (UTF8)
/usr/pgsql-15/share/pljava/pljava--unpackaged.sql as lines (UTF8)
/etc/sysconfig/pgsql/pljava.policy as lines (UTF8)
[root@dc-db-01-prod target]#
```

14. As you can make out from the above output, it has produced a list of files(.jar, .so, .sql) in “/usr/pgsql-15/share/pljava” and “/usr/pgsql-15/lib”:

15. find “libjvm” for jdk-17:

```
find / -name "*libjvm.so"
```

```
[root@dc-db-01-prod pljava]# find / -name "*libjvm.so*"
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.382.b05-2.el8.x86_64/jre/lib/amd64/server/libjvm.so
/usr/lib/jvm/java-17-openjdk-17.0.8.0.7-2.el8.x86_64/lib/server/libjvm.so
[root@dc-db-01-prod pljava]#
```

16. set both parameters in \$PGDATA/postgresql.conf and restart:

Ex.

```
dynamic_library_path = '/usr/pgsql-15/lib'
pljava.classpath = '/usr/pgsql-15/share/pljava/pljava-1.6.5.jar'
pljava.libjvm_location = '/usr/lib/jvm/java-17-openjdk-17.0.8.0.7-2.el8.x86_64/lib/server/libjvm.so'
# optional
pljava.statement_cache_size = 10
pljava.release_lingering_savepoints = true
pljava.vmoptions = '-Xmx128M -XX:+DisableAttachMechanism -Xms2m -XX:+UseSerialGC'
pljava.debug = false
```

17. Update your Java-17 security policy to allow the connection to your application (Optional):

```
find / -name "*java.policy"
```

```
/etc/sysconfig/pgsql/pljava.policy
/etc/java/java-1.8.0-openjdk/java-1.8.0-openjdk-1.8.0.382.b05-2.el8.x86_64/lib/security/java.policy
/etc/java/java-17-openjdk/java-17-openjdk-17.0.8.0.7-2.el8.x86_64/conf/security/java.policy
/etc/java/java-11-openjdk/java-11-openjdk-11.0.20.0.8-3.el8.x86_64/conf/security/java.policy_org
/etc/java/java-11-openjdk/java-11-openjdk-11.0.20.0.8-3.el8.x86_64/conf/security/java.policy
/root/pljava/pljava-1_6_5/pljava-packaging/src/main/resources/pljava.policy
/root/pljava/pljava-1_6_5/pljava-packaging/target/classes/pljava.policy
/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.382.b05-2.el8.x86_64/jre/lib/security/java.policy
/usr/java/jdk1.8.0_361-amd64/jre/lib/security/java.policy
```

```
vim /etc/java/java-17-openjdk/java-17-openjdk-17.0.8.0.7-2.el8.x86_64/conf/security/java.policy
```

Add this end of line and restart the PostgreSQL service.

```
grant {
    permission java.net.SocketPermission "APP_SVR_IP:PORT", "connect,resolve";
};
```

Ex.

```
grant {
    // allows anyone to listen on dynamic ports
    permission java.net.SocketPermission "localhost:0", "listen";

    // "standard" properties that can be read by anyone
    permission java.util.PropertyPermission "java.version", "read";
    permission java.util.PropertyPermission "java.vendor", "read";
    permission java.util.PropertyPermission "java.vendor.url", "read";
    permission java.util.PropertyPermission "java.class.version", "read";
    permission java.util.PropertyPermission "os.name", "read";
    permission java.util.PropertyPermission "os.version", "read";
    permission java.util.PropertyPermission "os.arch", "read";
    permission java.util.PropertyPermission "file.separator", "read";
    permission java.util.PropertyPermission "path.separator", "read";
    permission java.util.PropertyPermission "line.separator", "read";
    permission java.util.PropertyPermission "java.specification.version", "read";
    permission java.util.PropertyPermission "java.specification.maintenance.version", "read";
    permission java.util.PropertyPermission "java.specification.vendor", "read";
    permission java.util.PropertyPermission "java.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.specification.version", "read";
    permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
    permission java.util.PropertyPermission "java.vm.specification.name", "read";
    permission java.util.PropertyPermission "java.vm.version", "read";
    permission java.util.PropertyPermission "java.vm.vendor", "read";
    permission java.util.PropertyPermission "java.vm.name", "read";

    // Grant permission to connect to the IP address "10.27.16.10" on port "9000"
    permission java.net.SocketPermission "10.27.16.10:9000", "connect,resolve";
};
```

18. Now you can create pljava extension on sql prompt:

SQL; Create extension.

```
Create Pljava extension
CREATE EXTENSION pljava;
GRANT USAGE ON LANGUAGE java TO PUBLIC;
ALTER DEFAULT PRIVILEGES FOR ROLE postgres IN SCHEMA sqlj GRANT ALL ON TABLES TO PUBLIC;
GRANT ALL ON SCHEMA sqlj TO PUBLIC;
```

SQL; Drop extension.

```
DROP EXTENSION IF EXISTS pljava;
```

▪ Database Client Configurations

1. Install following packages to the below client server where you install GnuC apps:

```
dnf install postgresql15 postgresql15-server postgresql15-contrib postgresql15-odbc postgresql15-libs
dnf install unixODBC unixODBC-devel
```

2. Configure “odbcinst.ini” file:

```
vim /etc/odbcinst.ini
```

Comment or remove the existing configurations and put the below,

```
[PostgreSQL]
Description = ODBC for PostgreSQL
Driver      = /usr/pgsql-15/lib/psqlodbc.so
Setup       = /usr/lib64/libodbcpsqlS.so
Driver64    = /usr/pgsql-15/lib/psqlodbc.so
Setup64     = /usr/lib64/libodbcpsqlS.so
FileUsage   = 1
Threading   = 0
```

3. Configure “odbc.ini” file:

```
vim /etc/odbc.ini
```

```
[ODBC Data Sources]
PostgreSQL = PostgreSQL ODBC Driver
```

```
[DB_NAME]
Description = PostgreSQL Connection
Driver      = PostgreSQL
Trace       = Yes
TraceFile   = /var/log/trace.odbc.ini.log
Database    = DB_NAME
Servername  = DB_SVR_IP
UserName     = SCHEMA_NAME
Password    = SCHEMA_PW
Port        = DB_PORT
Protocol    = 9.5
ReadOnly    = No
RowVersioning = No
ShowSystemTables = No
ShowOidColumn = No
FakeOidIndex = No
ConnSettings =
```

4. Check the connection through command line whether DSN (Data Source Name) is correct as root user:

```
Isql -v <DB_NAME>
```