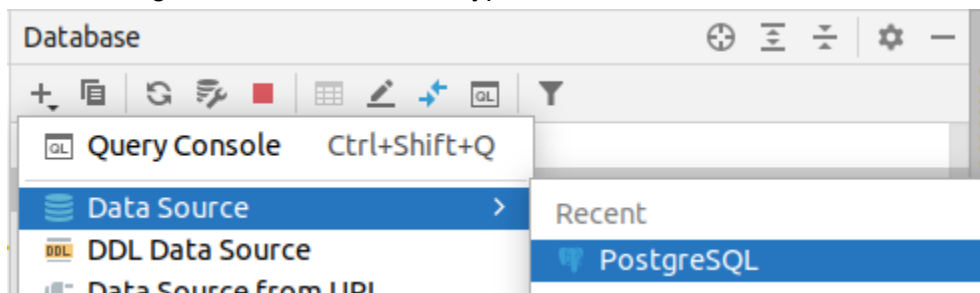


1. Log into AWS console
2. Go to the RDS dashboard
3. In the menu choose databases
4. Select a single database from the list
5. Copy "Endpoint" value somewhere
6. Go to EC2 dashboard
7. In menu, under "Network and security" select "Key pairs"
8. Click on "Create key pair"
9. Enter the name of the key pair and click "Create key pair"
10. The key pair was downloaded from the browser to your machine.
11. Generate public key from key pair. On Linux, you can do this by opening the terminal in the key pair folder and executing the command "**sudo chmod 400 key.pem**" to set appropriate access, and then you have to execute "**ssh-keygen -y -f key.pem > key.pub**". Here **key.pem** and **key.pub** are the names of your key, which can be different of course.
12. Go back to the EC2 dashboard
13. In the menu, under "Instances" select "Instances"
14. In instances list choose "bastion"
15. Copy "**Public IPv4 DNS**" somewhere
16. Click the "connect" button
17. Select "EC2 instance connect" and click "Connect"
18. EC2 instance terminal will open in the new window.
19. Type "**cd .ssh**" and press Enter
20. Type "**cat >> authorized\_keys**" and press Enter
21. Copy content of **key.pub** generated on step 11 to EC2 terminal, press enter, and Ctrl+C
22. Leave EC2 Terminal
23. Go to any tool which is used to set up DB connections (E.g. IntelliJ IDEA Ultimate)
24. Select PostgreSQL as connection type



## 25. Select “SSH/SSL”

**Data Sources and Drivers**

**Data Sources** Drivers

Project Data Sources

- postgres@localhost
- postgres@localhost [2]
- postgres@localhost [3]
- postgres@localhost [4]
- postgres@localhost [5]
- postgres@localhost [6]**
- postgres@wazpay-db-dev.cxef
- postgres@wazpay-db-prod.cet
- postgres@wazpay-db-test.cs7f

DDL Data Sources

- DDL data source

Problems 1

Name: postgres@localhost [6]

Comment:

General Options SSH/SSL Schemas Advanced

Connection type: default Driver: PostgreSQL

Host: localhost Port: 5432

Authentication: User & Password

User:

Password: <hidden> Save: Forever

Database: postgres

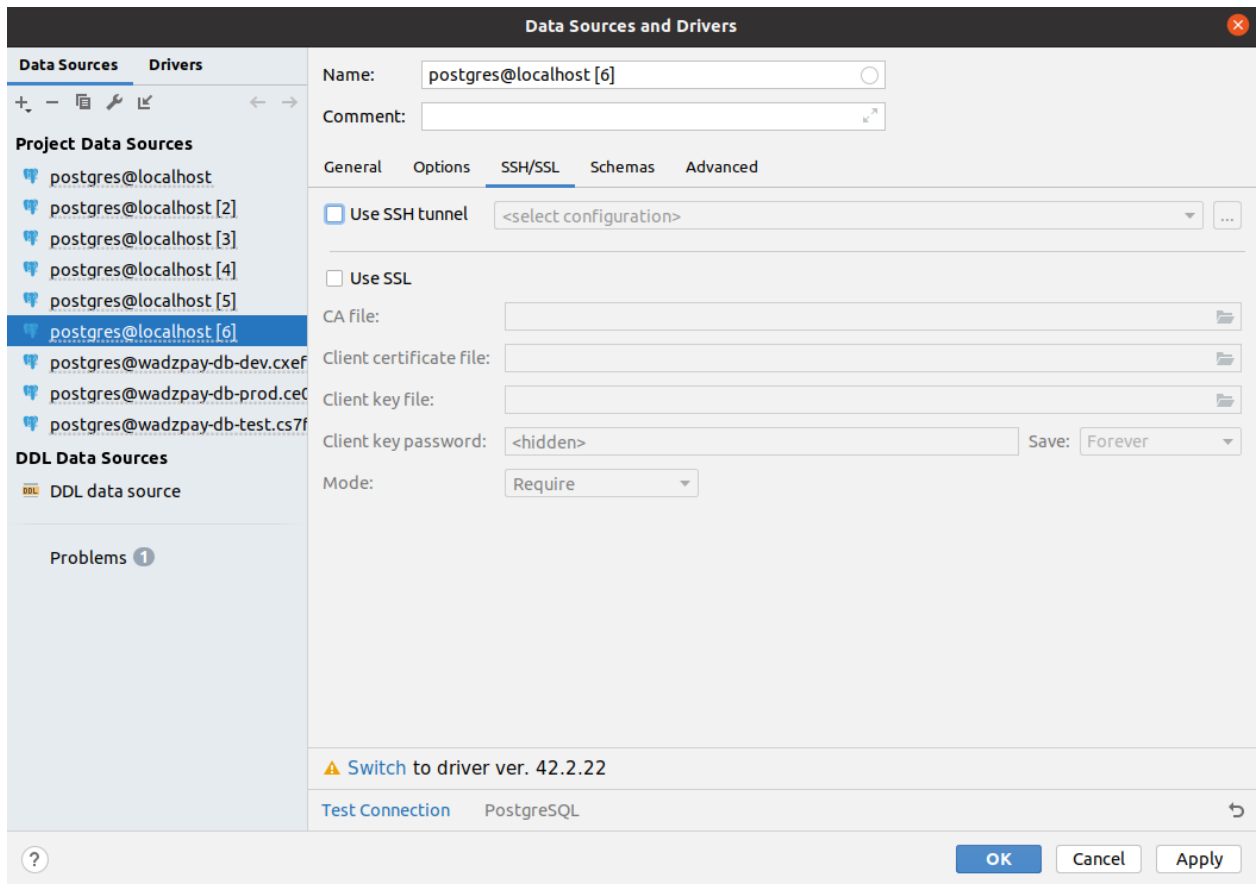
URL: jdbc:postgresql://localhost:5432/postgres  
Overrides settings above

Switch to driver ver. 42.2.22

Test Connection PostgreSQL

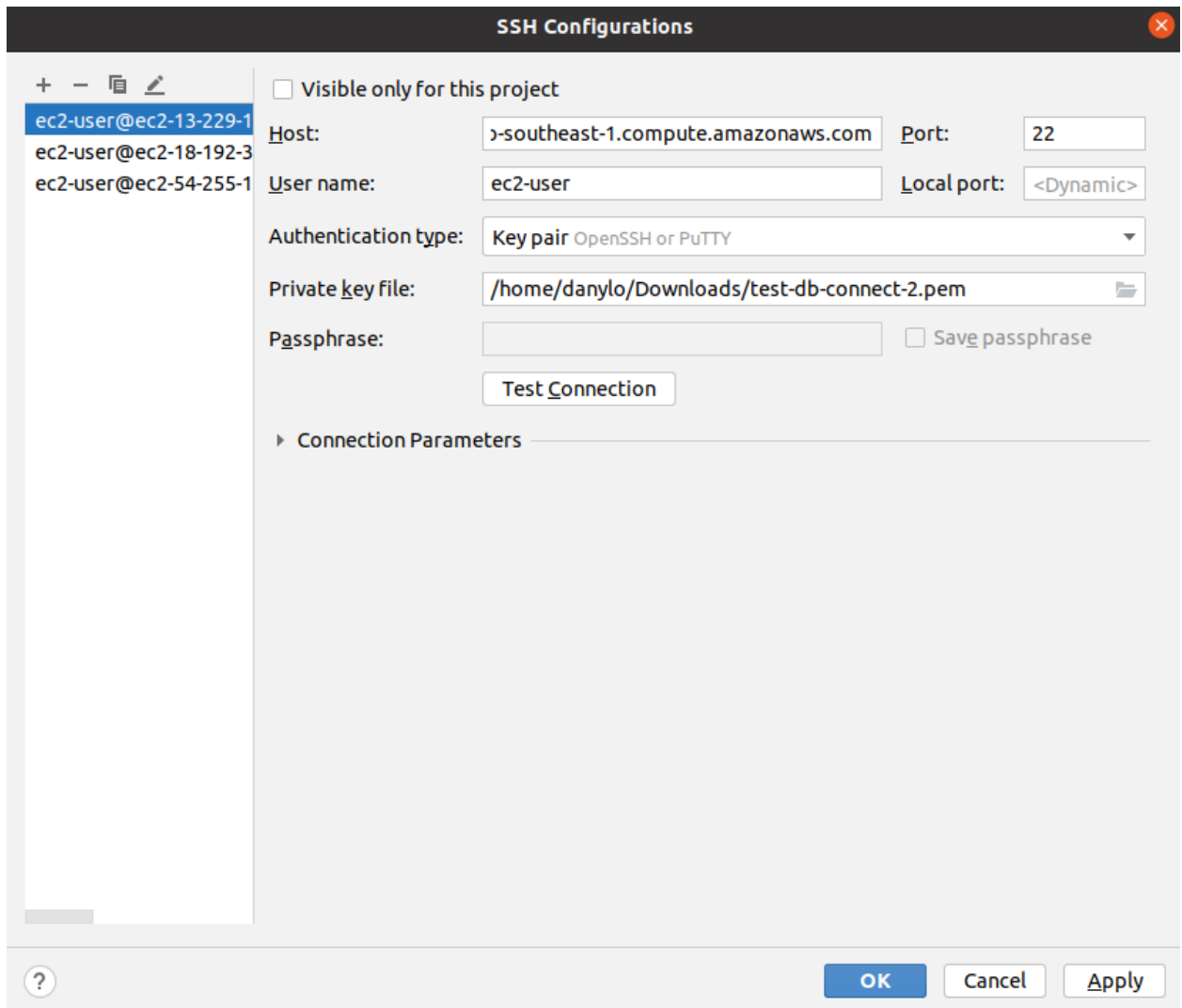
OK Cancel Apply

26. Click on “Use SSH tunnel checkbox”



27. Click on three dots

28. Press on +



The image shows a 'SSH Configurations' dialog box. On the left, there is a list of saved configurations: 'ec2-user@ec2-13-229-1', 'ec2-user@ec2-18-192-3', and 'ec2-user@ec2-54-255-1'. The first configuration is selected. To the right of the list, there is a checkbox labeled 'Visible only for this project'. Below this, the configuration details are shown: 'Host' is 's-southeast-1.compute.amazonaws.com', 'Port' is '22', 'User name' is 'ec2-user', and 'Local port' is '<Dynamic>'. The 'Authentication type' is set to 'Key pair OpenSSH or PuTTY'. The 'Private key file' is '/home/danylo/Downloads/test-db-connect-2.pem'. There is a 'Passphrase' field and a 'Save passphrase' checkbox. A 'Test Connection' button is located below the passphrase field. At the bottom, there is a 'Connection Parameters' section which is currently collapsed. The dialog has 'OK', 'Cancel', and 'Apply' buttons at the bottom right.

SSH Configurations

☐ Visible only for this project

Host: s-southeast-1.compute.amazonaws.com Port: 22

User name: ec2-user Local port: <Dynamic>

Authentication type: Key pair OpenSSH or PuTTY

Private key file: /home/danylo/Downloads/test-db-connect-2.pem

Passphrase:  ☐ Save passphrase

Test Connection

▶ Connection Parameters

OK Cancel Apply

29. For Host input value from step 15
30. For User name input "ec2-user"
31. Select Key pair authentication type.
32. In private key file use file downloaded at step 10
33. Test connection
34. Press OK
35. Go back from "SSH/SSL" to General
36. For host use the value from step 5
37. For user use "postgres"
38. For password contact Danylo
39. Test connection