



Hands-on Lab : Create Tables and Load Data in PostgreSQL using pgAdmin

Estimated time needed: 20 minutes

In this lab, you will learn how to create tables and load data in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool. The pgAdmin GUI provides an alternative to the command line for interacting with a PostgreSQL database using a graphical interface. This provides a number of key features for interacting with a PostgreSQL database in an easy to use format.

Software Used in this Lab

In this lab, you will use [PostgreSQL Database](#). PostgreSQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.



To complete this lab you will utilize the PostgreSQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

Database Used in this Lab

Books database has been used in this lab.

The following diagram shows the structure of the myauthors table from the Books database:

myauthors	
author_id	int
first_name	varchar(100)
middle_name	varchar(50)
last_name	varchar(100)

Objectives

After completing this lab, you will be able to use pgAdmin with PostgreSQL to:

- Create databases and tables in a PostgreSQL instance
- Load data into tables manually using the pgAdmin GUI
- Load data into tables from a text/script file

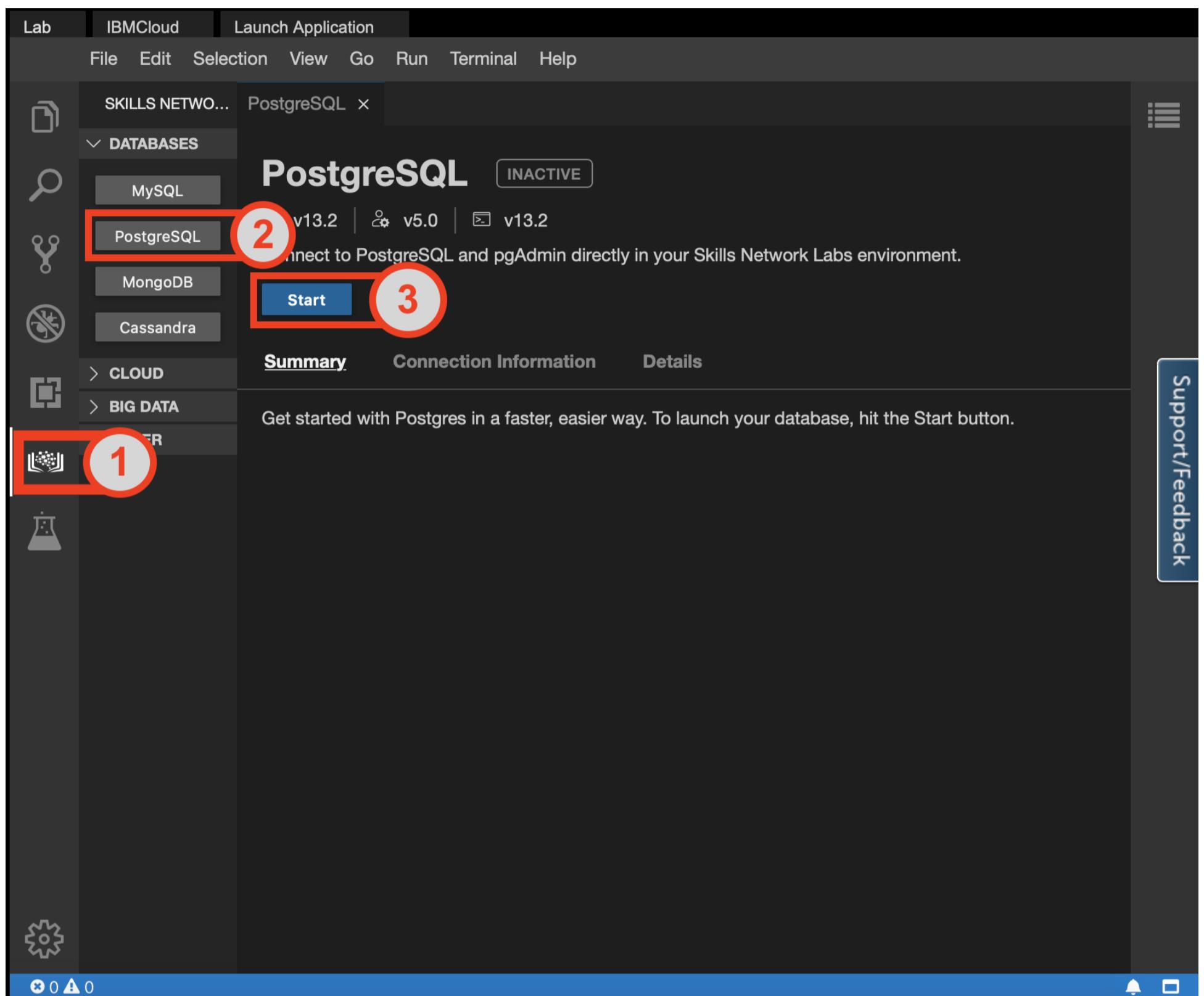
Lab Structure

In this lab, you will complete several tasks in which you will learn how to create tables and load data in the PostgreSQL database service using the pgAdmin graphical user interface (GUI) tool.

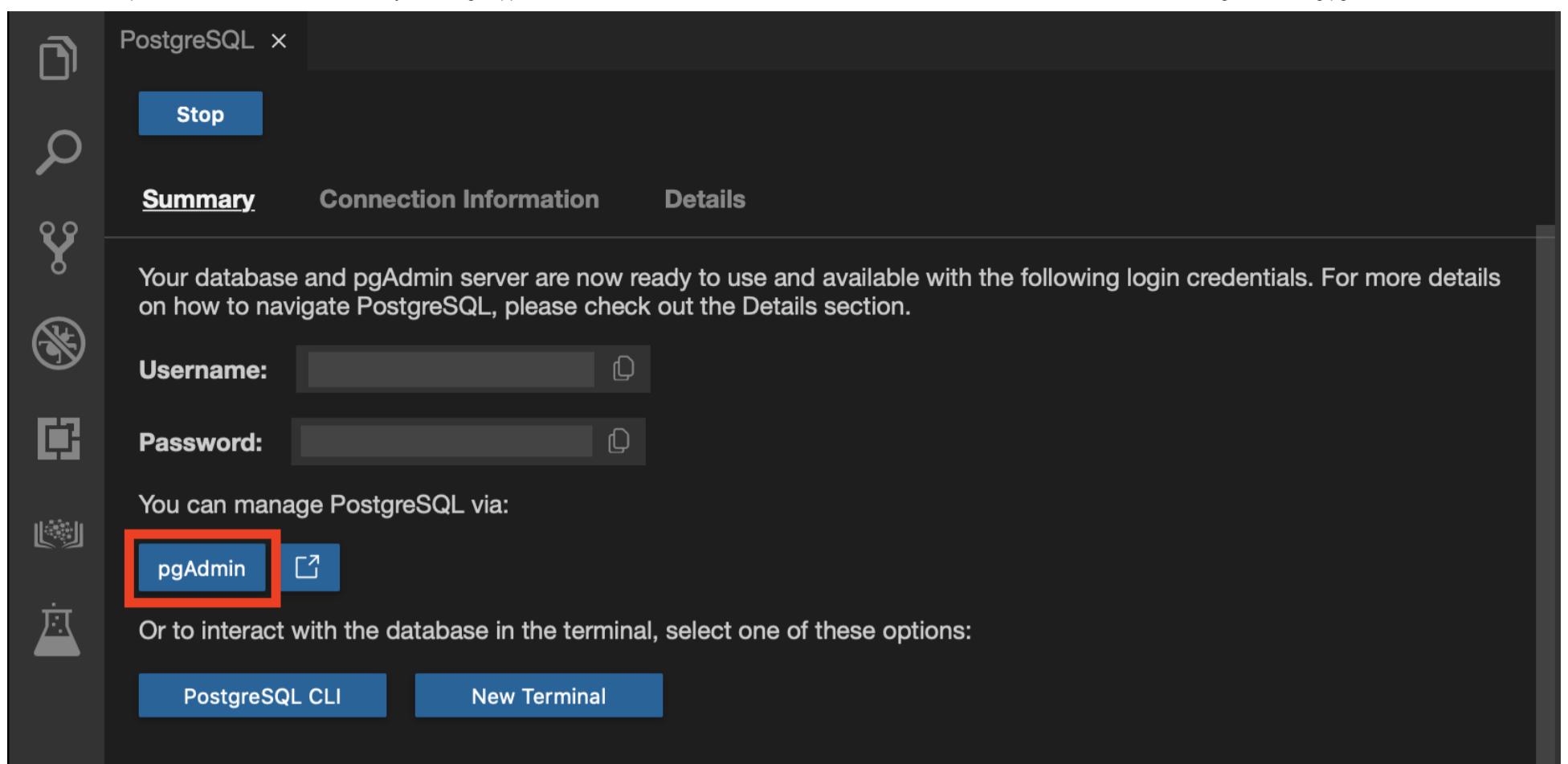
Task A: Create a database

First, to create a database on a PostgreSQL server instance, you'll first want to actually launch a PostgreSQL server instance on Cloud IDE and open up the pgAdmin Graphical User Interface.

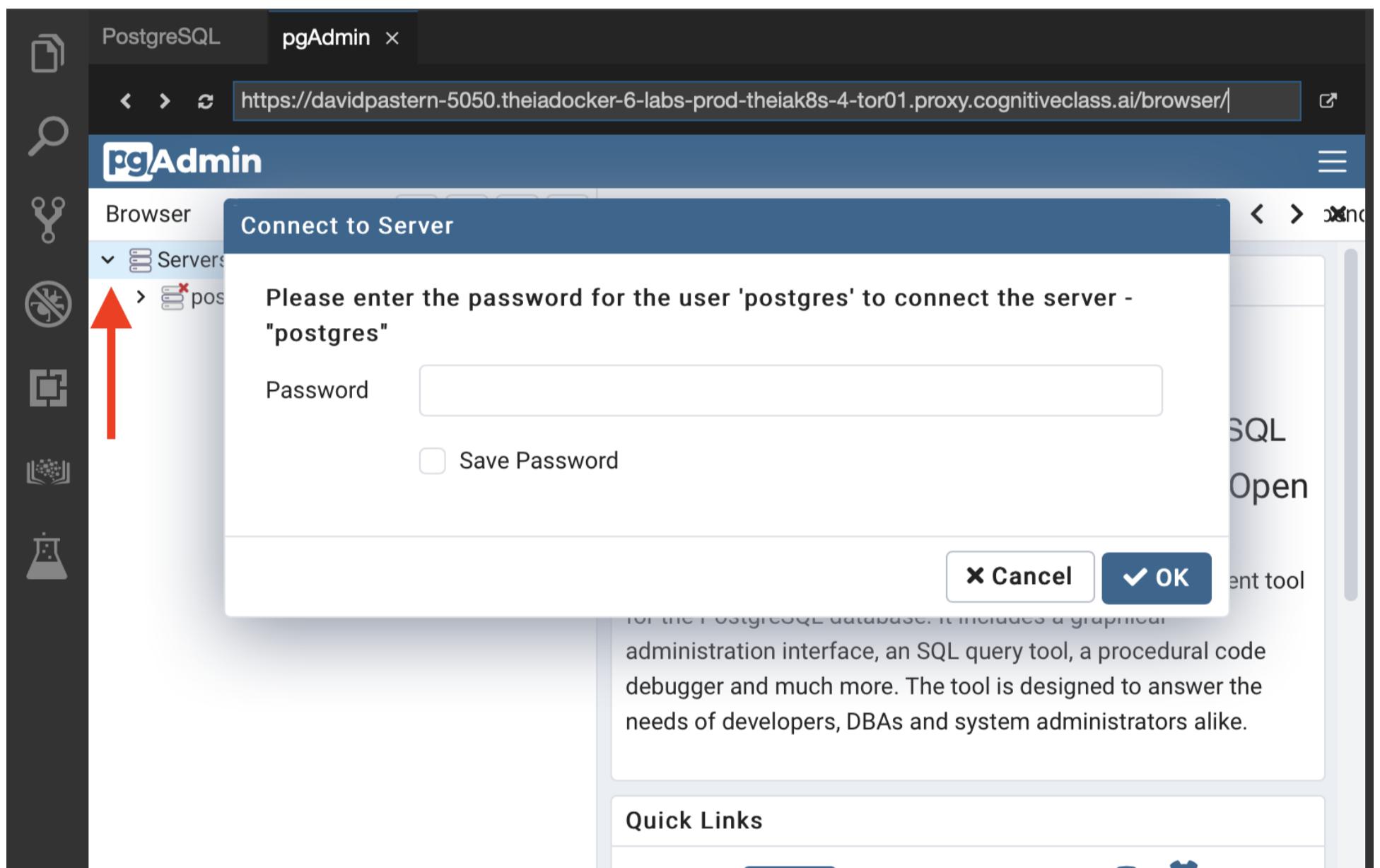
1. Click on the Skills Network extension button on the left side of the window.
2. Open the "DATABASES" drop down menu and click on "PostgreSQL"
3. Click on the "Start" button. PostgreSQL may take a few moments to start.



4. Next, open the pgAdmin Graphical User Interface by clicking the "pgAdmin" button in the Cloud IDE interface.



5. Once the pgAdmin GUI opens, click on the **Servers** tab on the left side of the page. You will be prompted to enter a password.



6. To retrieve your password, click on the "PostgreSQL" tab near the top of the interface.

7. Click on the Copy icon to the left of your password to copy the session password onto your clipboard.

PostgreSQL x pgAdmin

PostgreSQL

ACTIVE

v13.2 | v5.0 | v13.2

Connect to PostgreSQL and pgAdmin directly in your Skills Network Labs environment.

Stop

Summary **Connection Information** **Details**

Your database and pgAdmin server are now ready to use and available with the following login credentials. For more details on how to navigate PostgreSQL, please check out the Details section.

Username: [REDACTED]

Password: [REDACTED] **2**

You can manage PostgreSQL via:

pgAdmin

Or to interact with the database in the terminal, select one of these options:

PostgreSQL CLI **New Terminal**

8. Navigate back to the "pgAdmin" tab and paste in your password, then click **OK**

9. You will then be able to access the pgAdmin GUI tool.

← → C ⌘ ⌘ sandipsahajo-5050.theiadocker-27.proxy.cognitiveclass.ai/browser/

pgAdmin File Object Tools Help

Browser Dashboard Properties SQL Statistics Dependencies Dependents

Welcome

pgAdmin
Management Tools for PostgreSQL

Feature rich | Maximises PostgreSQL | Open Source

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 developers, DBAs and system administrators alike.

Quick Links

Add New Server

Getting Started

PostgreSQL Documentation **pgAdmin Website** **Planet PostgreSQL** **Community Support**

10. In the tree-view, expand **Servers > postgres > Databases**. If prompted, enter your PostgreSQL service session password. Right-click on **Databases** and go to **Create > Database**. In the **Database** box, type **Books** as the name for your new database, and then click **Save**. Proceed to Task B.

The screenshot shows the pgAdmin interface with the following steps highlighted:

- Browser tab (1)
- Servers (1) node (2)
- Databases (1) node (3)

A context menu is open over the 'postgres' database node, with the 'Create' option selected. A sub-menu is displayed with the 'Database...' option highlighted and a cursor icon pointing at it.

The 'Create - Database' dialog box is open, showing the 'General' tab selected. The 'Database' field contains the value 'Books'. The 'Owner' field is set to 'postgres'. The 'Comment' field is empty.

At the bottom of the dialog, there are three buttons: 'Cancel', 'Reset', and 'Save'. The 'Save' button is highlighted with a red border and a cursor icon pointing at it.

Task B: Create tables

Now that you have your PostgreSQL service active and have created the **Books** database using pgAdmin, let's go ahead and create a few tables to populate the database and store the data that we wish to eventually upload into it.

1. In the tree-view, expand **Books** > **Schemas** > **public**. Right-click on **Tables** and go to **Create** > **Table**.

The screenshot shows the pgAdmin interface with the following details:

- Browser Panel:** Shows the database structure. A path is highlighted with red boxes and numbers:
 - 1. Databases (2) > Books
 - 2. Schemas (1) > public
 - 3. Tables
 - 4. Create > Table...
- Dashboard Panel:** Shows Database sessions (1), Tuples in (1), and Server activity.
- Context Menu (Open in sub-panel):** Options include Refresh..., Grant Wizard..., Search Objects..., and Query Tool.

2. On the **General** tab, in the **Name** box, type **myauthors** as name of the table. Don't click Save, proceed to the next step.

Create - Table

X

General Columns Advanced Constraints Partitions Parameters Security SQL

Name myauthors

Owner postgres

Schema public

Tablespace Select an item...

Partitioned table? No

Comment

i ?

Cancel Reset Save

3. Switch to tab **Columns** and click the **Add new row** button four times to add **4** column placeholders. Don't click Save, proceed to the next step.

Create - Table

General **Columns** Advanced Constraints Partitions Parameters Security SQL

Inherited from table(s) Select to inherit from...

Columns

	Name ▾	Data type	Length/Precision	Scale	Not NULL?	Primary key?
		<input type="text"/>	Select an item... ▾		<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
		<input type="text"/>	Select an item... ▾		<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
		<input type="text"/>	Select an item... ▾		<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No
		<input type="text"/>	Select an item... ▾		<input checked="" type="checkbox"/> No	<input checked="" type="checkbox"/> No

Cancel Reset Save

4. Enter the **myauthors** table definition structure information as shown in the image below in the highlighted boxes. Then click **Save**.
Proceed to Task C.

The screenshot shows the 'Create - Table' dialog in pgAdmin. The 'Columns' tab is selected. A table is displayed with columns for Name, Data type, Length/Precision, Scale, Not NULL?, and Primary key?. The 'author_id' column is highlighted with a red box. The 'Save' button at the bottom right is also highlighted with a red box.

	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
<input type="button" value="Edit"/> <input type="button" value="Delete"/>	author_id	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input type="button" value="Edit"/> <input type="button" value="Delete"/>	first_name	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input type="button" value="Edit"/> <input type="button" value="Delete"/>	middle_name	character varying	50		<input type="checkbox"/> No	<input type="checkbox"/> No
<input type="button" value="Edit"/> <input type="button" value="Delete"/>	last_name	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No

Task C: Load data into tables manually using the pgAdmin GUI

Great! You now have a database and have created tables within it. With the pgAdmin GUI, you can insert values into the tables manually. This is useful if you have a few new entries you wish to add to the database. Let's see how to do it.

1. In the tree-view, expand **Tables**. Right-click on **myauthors** and go to **View/Edit Data > All Rows**.

The screenshot shows the pgAdmin interface with the following details:

- Top Bar:** pgAdmin, File, Object, Tools, Help.
- Toolbar:** Browser, Servers, Databases, Books, Catalogs, Event Triggers, Extensions, Foreign Data Wrap, Languages, Publications, Schemas, Collations, Domains, FTS Config, FTS Dictionary, FTS Parser, FTS Template, Foreign Table, Functions, Materialized View, Procedures, Sequences, Tables, Columns, Constraints, Indexes, RLS Policies, Rules, Triggers.
- Servers List:** Servers (1) > postgres > Databases (2) > Books > Casts, Catalogs, Event Triggers, Extensions, Foreign Data Wrap, Languages, Publications, Schemas (1) > public > Collations, Domains, FTS Config, FTS Dictionary, FTS Parser, FTS Template, Foreign Table, Functions, Materialized View, Procedures, Sequences, Tables (1).
- Context Menu (Open at Tables (1)):** Create, Refresh..., Count Rows, Delete/Drop, Drop Cascade, Reset Statistics, Import/Export..., Maintenance..., Scripts, Truncate, Backup..., Restore..., View/Edit Data, Search Objects..., Query Tool, Properties... (highlighted in blue).
 - View/Edit Data Submenu:** All Rows (selected), First 100 Rows, Last 100 Rows, Filtered Rows...
- Table Selection:** Tables (1) > myauthors (highlighted with a red box and labeled '2').
- Table Structure:** myauthors > Columns, Constraints (1), Indexes, RLS Policies, Rules, Triggers.

2. You will insert 2 rows of data into the **myauthors** table. In the lower **Data Output** pane, enter **myauthors** table data information for 2 rows as shown in the highlighted boxes in the image below. Then click the **Save Data Changes** button. Proceed to Task D.

The screenshot shows the pgAdmin interface. At the top, there's a navigation bar with tabs: Dashboard, Properties, SQL, Statistics, Dependencies, Dependents, and a selected tab labeled 'public.myauthors/Books'. Below the navigation bar is a toolbar with various icons, one of which is highlighted with a red box and a cursor icon. A large red arrow points from this icon to a red oval containing the text 'Save Data Changes icon'. The main area contains a 'Query Editor' tab with a SELECT query and a 'Data Output' tab showing a table with two rows of data. The first row has author_id 1, first_name Merrit, middle_name [null], and last_name Eric. The second row has author_id 2, first_name Linda, middle_name [null], and last_name Mul.

	author_id [PK] integer	first_name character varying (100)	middle_name character varying (50)	last_name character varying (100)
1	1	Merrit	[null]	Eric
2	2	Linda	[null]	Mul

Task D: Load data into tables using a text/script file

In the previous task, you entered some data entries into a table manually with pgAdmin. While this method can be useful for small additions, if you wish to upload large amounts of data at once, that process becomes far too tedious. An alternative is to load data into tables from a text or script file containing the data you wish to enter. Let's take a look at how to do this.

- Finally, you will import the remainder of the **myauthors** table data from a csv text file. Download the csv file below to your local computer:
 - [myauthors.csv](#)
- In the tree-view, right-click on **myauthors** and go to **Import/Export**.

The screenshot shows the pgAdmin interface with a dark blue header bar containing the pgAdmin logo and menu items: File, Object, Tools, and Help. Below the header is a toolbar with icons for Browser, Dashboard, and other functions. The main area features a tree view on the left under the 'Servers' node, showing 'Servers (1)' and 'postgres'. The 'postgres' node is expanded, showing 'public' and 'myauthors' databases. The 'myauthors' database node is selected, indicated by a blue border.

The screenshot shows the pgAdmin interface with the following details:

- Left Panel (Object Navigator):** Shows the database structure:
 - Databases (2): Books, public
 - Books: Casts, Catalogs, Event Triggers, Extensions, Foreign Data W, Languages, Publications, Schemas (1)
 - Schemas (1): public
 - Collation, Domain, FTS Co, FTS Dic, FTS Par, FTS Ter, Foreign, Function, Material, Procedure, Sequence, Tables, mya
- Center Panel (Query Editor):** Displays a context menu for the 'public' schema, with 'Import/Export...' highlighted.
- Bottom Panel (Properties):** Shows a table with one row and two columns, with the value '2' in the second column.

Two red numbers are overlaid on the left panel:

- 1** is next to the 'Tables' node under the 'public' schema.
- 2** is next to the 'maya' node under the 'public' schema.

3. Follow the instructions below to import:

- Make sure Import/Export is set to **Import**, Format = **csv** and Header = **Yes**. Then click on the **Select file** button by the Filename box.

Import/Export data - table 'myauthors'

Options Columns

Import/Export **Import** **1**

File Info **4**

Filename **...**

Format **csv** **2**

Encoding **Select an item...**

Miscellaneous

OID **No**

Header **Yes** **3**

Delimiter **Select from list...**

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

Cancel **OK**

- Click the **Upload File** button.

Select file

/var/lib/pgadmin/

Name	Size	Modified	Upload File
sessions	4.0 kB	Mon Mar 22 02:15:08 2021	
storage	4.0 kB	Mon Mar 22 02:11:24 2021	

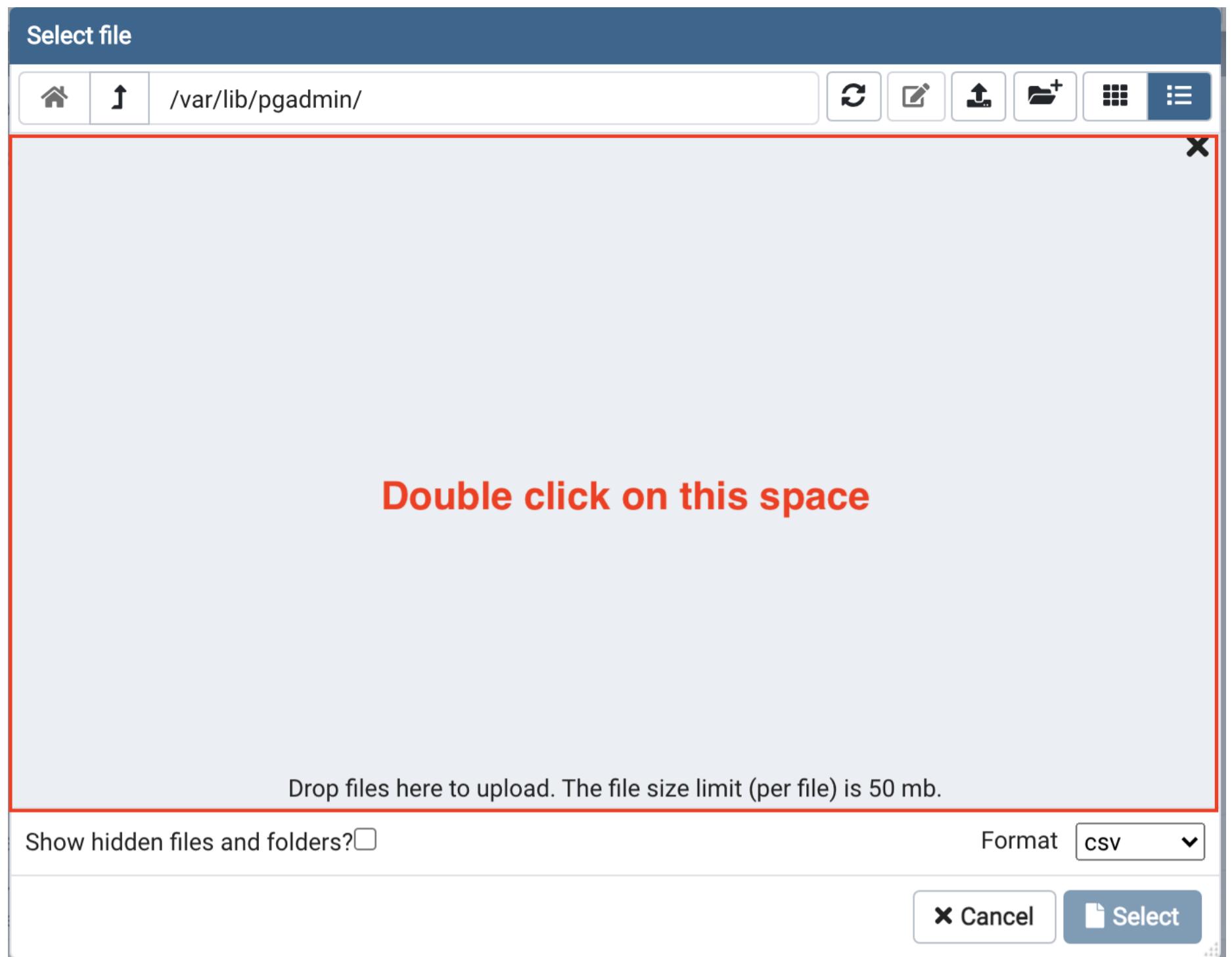
Show hidden files and folders?

Format **csv**

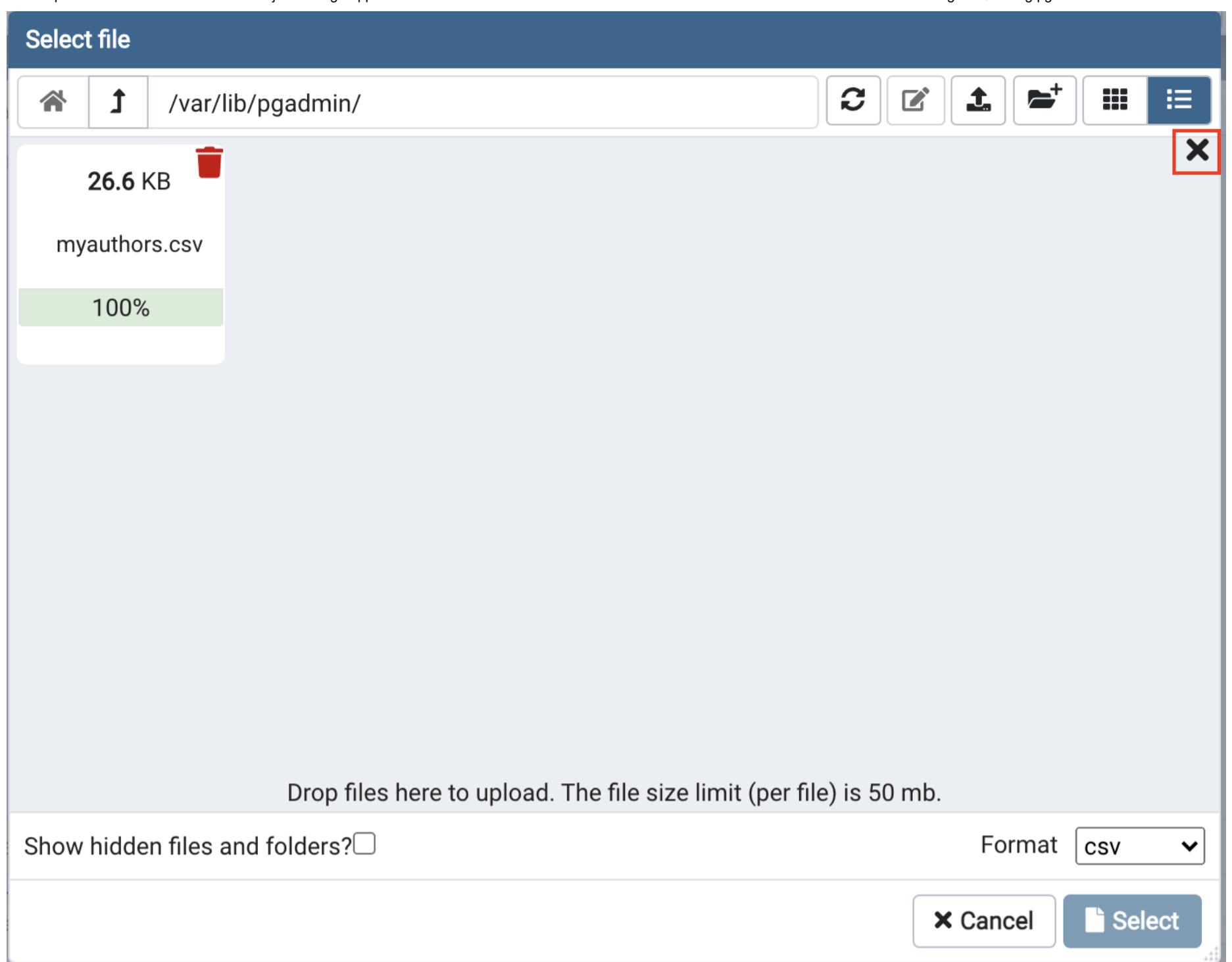
Cancel

Select

- o Double-click on the drop files area and load the **myauthors.csv** you downloaded earlier from your local computer storage.



- o When the upload is complete, close the drop files area clicking the X button.



- Select the uploaded **myauthors.csv** file from the list and click the **Select** button.

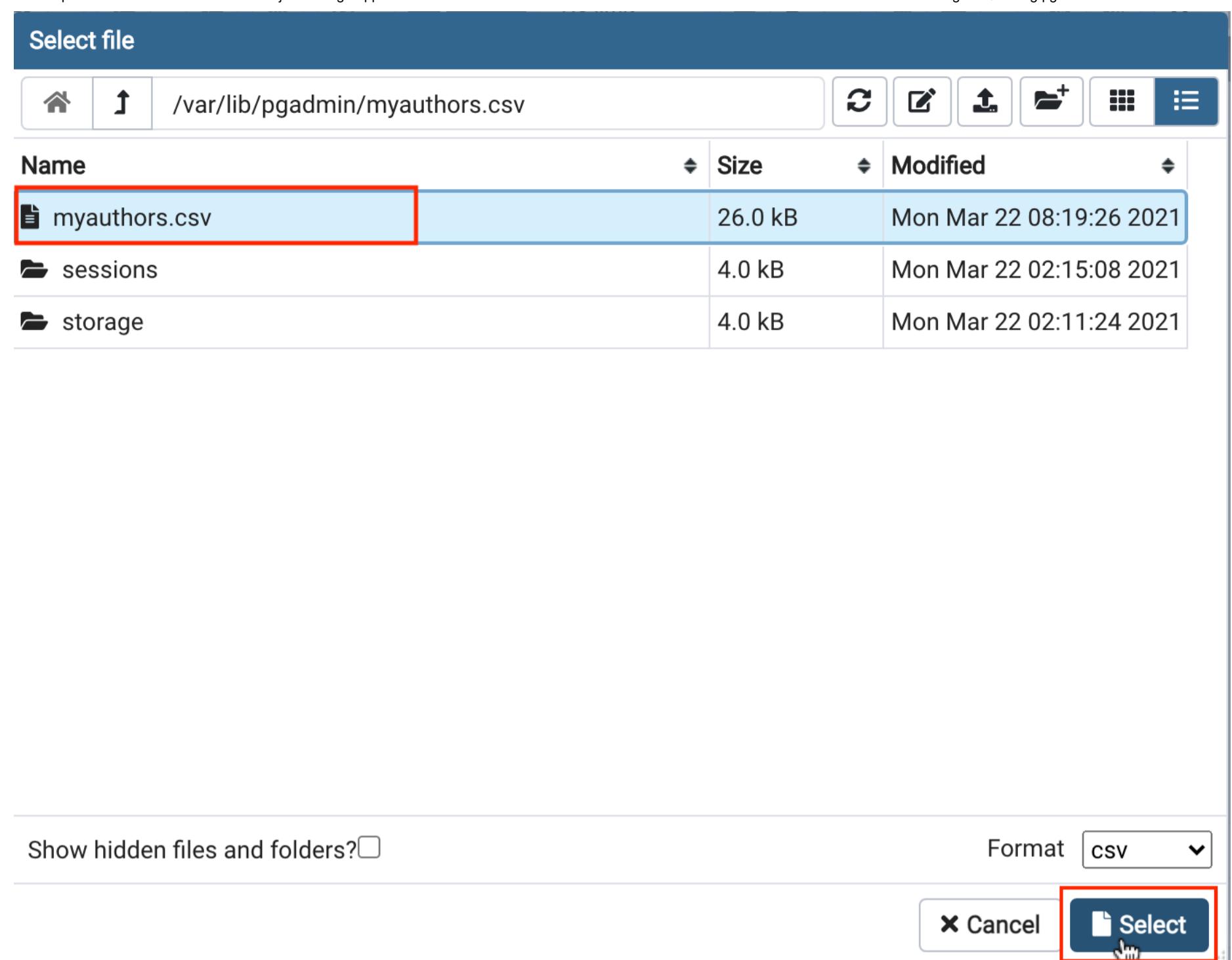
Select file

/var/lib/pgadmin/myauthors.csv

Name	Size	Modified
myauthors.csv	26.0 kB	Mon Mar 22 08:19:26 2021
sessions	4.0 kB	Mon Mar 22 02:15:08 2021
storage	4.0 kB	Mon Mar 22 02:11:24 2021

Show hidden files and folders?

Format



- Click **OK** and notification of import success should appear.

Import/Export data - table 'myauthors'

Options Columns

Import/Export Import

File Info

Filename ...

Format

Encoding

Miscellaneous

OID No

Header Yes

Delimiter ▼

Specifies the character that separates columns within each row (line) of the file. The default is a tab character in text format, a comma in CSV format. This must be a single one-byte character. This option is not allowed when using binary format.

✗ Cancel ✓ OK

Import - Copying table data

Copying table data 'public.myauthors' on database 'Books' and server (postgres:5432)

Mon Mar 22 2021 02:26:40 GMT-0600 (Mountain Daylight Time)

⌚ 0.02 seconds More details... ✗ Stop Process

✓ Successfully completed.

4. Repeat Task C Step 1 to check that the newly imported data rows appear along with your previously inserted 2 rows.

Dashboard Properties SQL Statistics Dependencies Dependents public.myauthors/Books/postgres@postgres

public.myauthors/Books/postgres@postgres

Query Editor Query History

```
1 SELECT * FROM public.myauthors
2 ORDER BY author_id ASC
```

Data Output Explain Messages Notifications

	author_id [PK] integer	first_name character varying (100)	middle_name character varying (50)	last_name character varying (100)
1	1	Merrit	[null]	Eric
2	2	Linda	[null]	Mul
3	3	Alecos	[null]	Papadatos
4	4	Paul	C.van	Oorschot
5	5	David	[null]	Cronin
6	6	Richard	[null]	Blum
7	7	Yuval	Noah	Harari
8	8	Paul	[null]	Albitz
9	9	David	[null]	Beazley
10	10	John	Paul	Shen
11	11	Andrew	[null]	Miller
12	12	Melanie	[null]	Swan
13	13	Neal	[null]	Ford
14	14	Nir	[null]	Shavit
15	15	Tim	[null]	Kindberg
16	16	Mike	[null]	McQuaid
17	17	Brian	P.	Hogan
18	18	Jean-Philippe	[null]	Aumasson
19	19	Lance	[null]	Fortnow
20	20	Richard	C.	Jeffrey
21	21	William	L.	Simon
22	22	Magnus	Lie	Hetland
23	23	Mike	[null]	McShaffry
24	24	Norman	[null]	Matloff
25	25	John	E.	Hopcroft
26	26	S.	[null]	Sudarshan

As you can see, the data contained in the **csv** file was successfully uploaded into the table and you did not have to manually input hundreds of entries.

Conclusion

Congratulations! You have completed this lab, and you are ready for the next topic.

Author

- [Sandip Saha Joy](#)

Other Contributors

- [David Pasternak](#)

Changelog

Date	Version	Changed by	Change Description
2021-03-15	1.0	Sandip Saha Joy	Created initial version
2021-10-18	1.1	David Pasternak	Updated lab instructions

© IBM Corporation 2021. All rights reserved.