Guided Exercise: Build Developer Environments with Compose

Configure a repeatable developer environment with Podman Compose.

Outcomes

You should be able to:

- Create a compose file that contains the definition of a PostgreSQL server and a pgAdmin interface.
- Create a compose file that contains the definition of a pgAdmin interface.
- Start and run the developer environment.
- Access the pgAdmin interface from a web browser to retrieve the data from the tables.

As the `student` user on the `workstation` machine, use the `lab` command to prepare your system for this exercise.

This command copies the necessary files for the development environment.

```
[student@workstation ~]$ lab start compose-environments
```

Instructions

1. Create a compose file that contains the definition of a PostgreSQL server.

   Change to the `/home/student/DO188/labs/compose-environments` directory and open the `compose.yml` file.

   ```
   [student@workstation ~]$ cd ~/DO188/labs/compose-environments
   [student@workstation compose-environments]$ gedit compose.yml
   ```

   Define a database container that uses the `registry.ocp4.example.com:8443/rhel9/postgresql-13:1` image. Forward port `5432` from the localhost to the same port inside the container.

   ```
   services:
     db:
       image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
       ports:
         - "5432:5432"
   ```

   Define the following environment variables:

Field	Value
POSTGRESQL_USER	`backend`
POSTGRESQL_DATABASE	`rpi-store`
POSTGRESQL_PASSWORD	`redhat`

   ```
   services:
     db:
       image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
       environment:
         POSTGRESQL_USER: backend
         POSTGRESQL_DATABASE: rpi-store
         POSTGRESQL_PASSWORD: redhat
       ports:
         - "5432:5432"
   ```

   Bind mount the `/home/student/DO188/labs/compose-environments/database_scripts` directory to the `/opt/app-root/src/postgresql-start` directory with the `z` option for SELinux. You can use the relative path to the `compose.yml` file for the `database_scripts` directory as the bind mount.

```
services:
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
```

Define a persistent volume called `rpi` for the container. Bind mount the `rpi` volume to the `/var/lib/pgsql/data` directory in the container.

```
services:
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
      - rpi:/var/lib/pgsql/data

volumes:
  rpi: {}
```

Call the container `compose_environments_postgresql`, and save the file.

```
services:
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    container_name: "compose_environments_postgresql"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
      - rpi:/var/lib/pgsql/data
volumes:
  rpi: {}
```

2. Define a pgAdmin server in the `compose.yml` file.

   Define a database admin interface container that uses the `registry.ocp4.example.com:8443/crunchydata/crunchy-pgadmin4:ubi8-4.30-1` image. Map port `5050` from the container to port `5050` on the host.

```
services:
  db-admin:
    image: "registry.ocp4.example.com:8443/crunchydata/crunchy-pgadmin4:ubi8-4.30-1"
    ports:
      - "5050:5050"
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    container_name: "compose_environments_postgresql"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
      - rpi:/var/lib/pgsql/data

volumes:
  rpi: {}
```

Define the following environment variables:

Field	Value
PGADMIN_SETUP_EMAIL	`user@example.com`
PGADMIN_SETUP_PASSWORD	`redhat`

```
services:
  db-admin:
    image: "registry.ocp4.example.com:8443/crunchydata/crunchy-pgadmin4:ubi8-4.30-1"
    environment:
      PGADMIN_SETUP_EMAIL: user@example.com
      PGADMIN_SETUP_PASSWORD: redhat
    ports:
      - "5050:5050"
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    container_name: "compose_environments_postgresql"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
      - rpi:/var/lib/pgsql/data

volumes:
  rpi: {}
```

Name the container `compose_environments_pgadmin`. Save and close the file.

```
services:
  db-admin:
    image: "registry.ocp4.example.com:8443/crunchydata/crunchy-pgadmin4:ubi8-4.30-1"
    container_name: "compose_environments_pgadmin"
    environment:
      PGADMIN_SETUP_EMAIL: user@example.com
      PGADMIN_SETUP_PASSWORD: redhat
    ports:
      - "5050:5050"
  db:
    image: "registry.ocp4.example.com:8443/rhel9/postgresql-13:1"
    container_name: "compose_environments_postgresql"
    environment:
      POSTGRESQL_USER: backend
      POSTGRESQL_DATABASE: rpi-store
      POSTGRESQL_PASSWORD: redhat
    ports:
      - "5432:5432"
    volumes:
      - ./database_scripts:/opt/app-root/src/postgresql-start:Z
      - rpi:/var/lib/pgsql/data

volumes:
  rpi: {}
```

NOTE

You can refer to the completed `compose.yml` file in the `/home/student/DO188/solutions/compose-environments` directory.

3. Run the developer environment.

   From the `/home/student/DO188/labs/compose-environments` directory, use the `compose.yml` file to start the containerized development environment. Use the `-d` option to run the containers in the background.

```
[student@workstation compose-environments]$ podman compose up -d
['podman', '--version', '']
using podman version: ...
** excluding:  set()
['podman', 'network', 'exists', 'compose-environments_default']
...output omitted...
exit code: 0
```

Confirm that the two containers are running.

```
[student@workstation compose-environments]$ podman compose ps
using podman version: ...
podman ps -a --filter label=io.podman.compose.project=compose-environments
CONTAINER ID IMAGE       COMMAND          CREATED    STATUS  PORTS      NAMES
d64b...5c6f  registry... /opt/crunchy... 23 sec...  Up...   ...5050... ...pgadmin
91ae...474e  registry... run-postgresql  23 sec...  Up...   ...5432... ...postg...
exit code: 0
```

Confirm that the persistent volumes exist.

```
[student@workstation compose-environments]$ podman volume list
DRIVER     VOLUME NAME
local      91a6...f45d
local      bd15...a5e2
local      compose-environments_rpi
local      f056...7eb0
```

NOTE

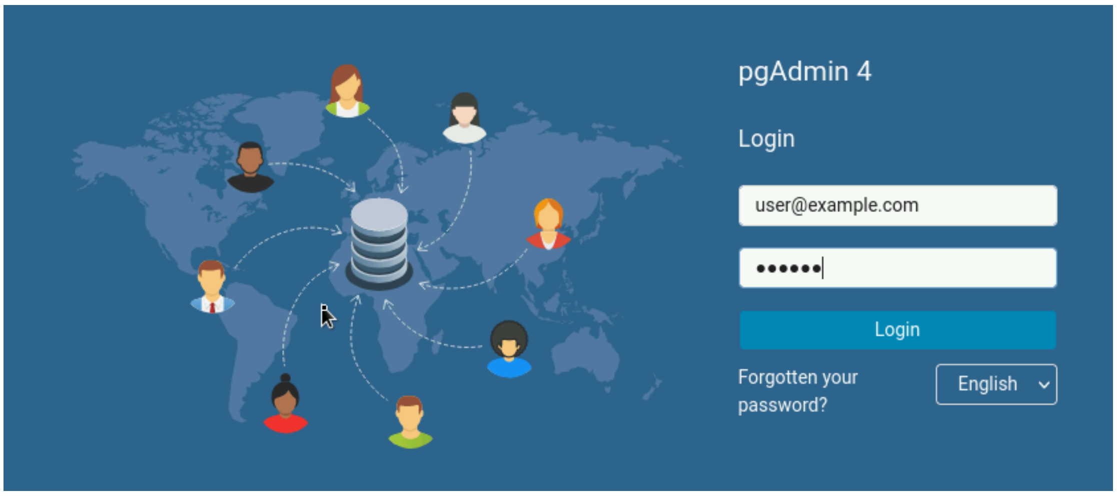
The command might display additional volumes from previous exercises.

Retrieve the logs from both containers and confirm that errors are not reported in the logs. Press **Ctrl+C** to exit the logs.

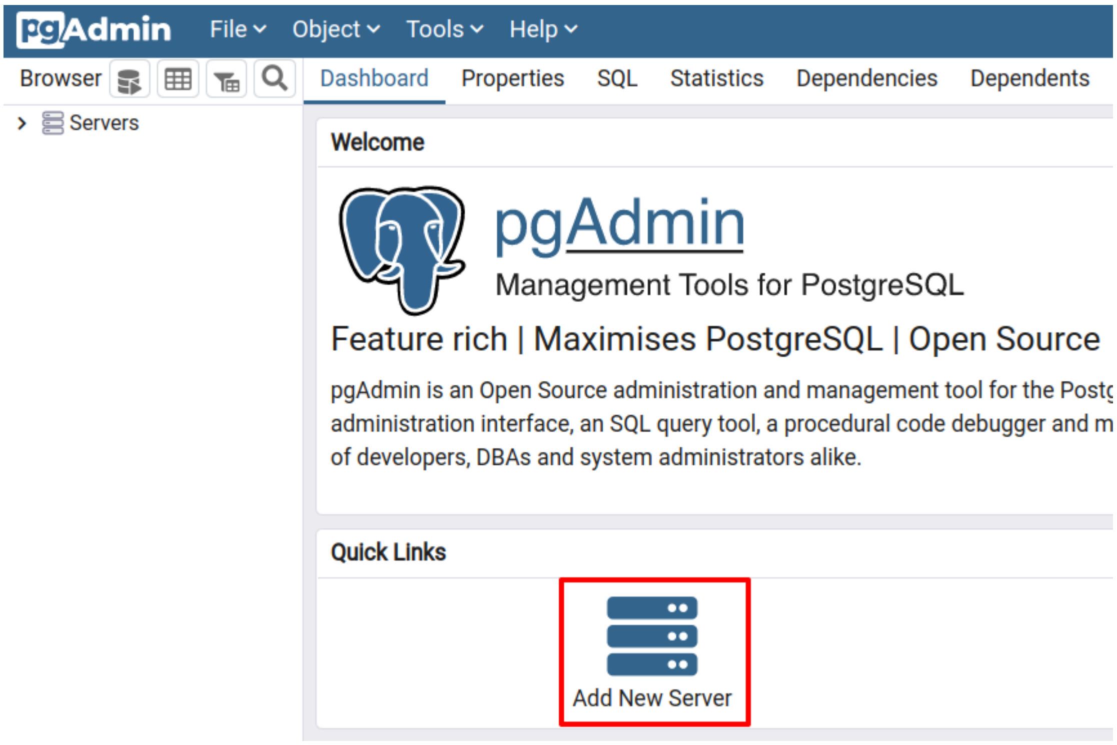
```
[student@workstation compose-environments]$ podman compose logs -n -f
['podman', '--version', '']
using podman version: ...
podman logs -f -n compose_environments_pgadmin compose_environments_postgresql
...output omitted...
compose_environments_postgresql Starting server...
compose_environments_postgresql 2022-08-11 14:42:02.237 UTC [1] LOG:  redirecting log output to logging collector process
compose_environments_postgresql 2022-08-11 14:42:02.237 UTC [1] HINT:  Future log output will appear in directory "log".
...output omitted...
compose_environments_pgadmin Thu Aug 11 14:41:57 UTC 2022 INFO: Setting up pgAdmin4 database..
compose_environments_pgadmin Thu Aug 11 14:42:05 UTC 2022 INFO: Starting Apache web server..
```

4. Access the pgAdmin interface from a web browser. Retrieve and modify data from the database.

   Open a web browser and go to `http://localhost:5050`. Access the pgAdmin interface as the `user@example.com` user with the `redhat` password.



   Click **Add New Server** to connect to the `compose_environments_postgresql` database container.



pgAdmin
Management Tools for PostgreSQL
Feature rich | Maximises PostgreSQL | Open Source
pgAdmin is an Open Source administration and management tool for the Postg
administration interface, an SQL query tool, a procedural code debugger and m
of developers, DBAs and system administrators alike.

On the `General` tab, set `rpi-store` as the name.

Switch to the `Connection` tab. Complete the form with the following data and leave the rest of the fields with their default values.

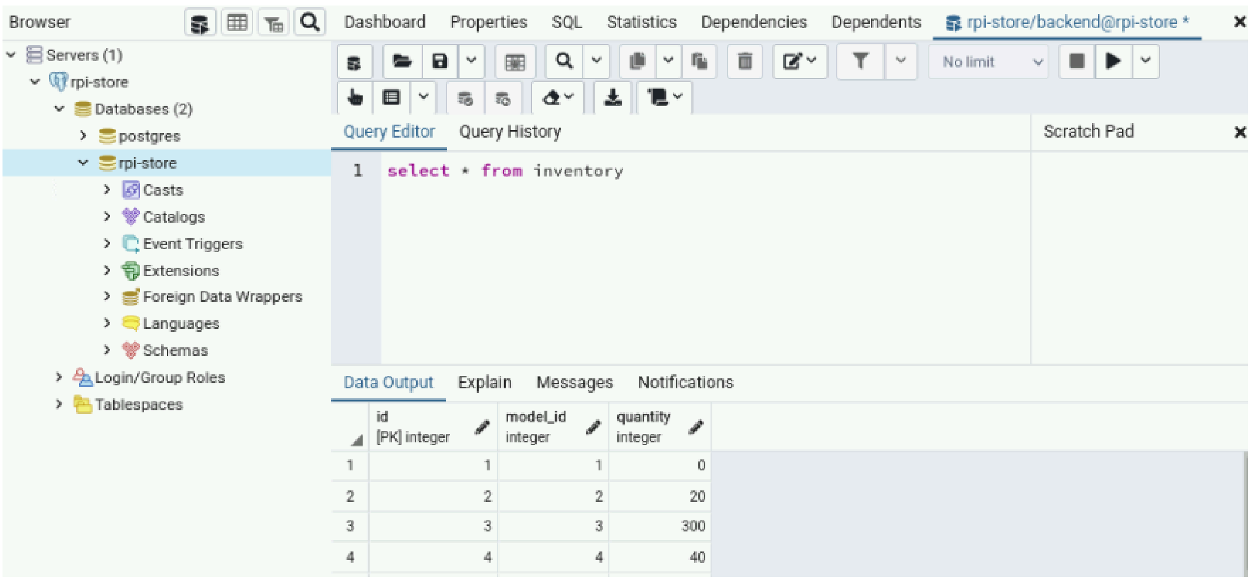
Field	Value
Host name/address	`db`
Username	`backend`
Password	`redhat`

Click **Save**. The application verifies the connection before exiting the form.

Go to **Servers** → **rpi-store** → **Databases** → **rpi-store**, and then select **Tools** → **Query Tool** from the menu. In the **Query Editor**, enter the following query.

```
select * from inventory
```

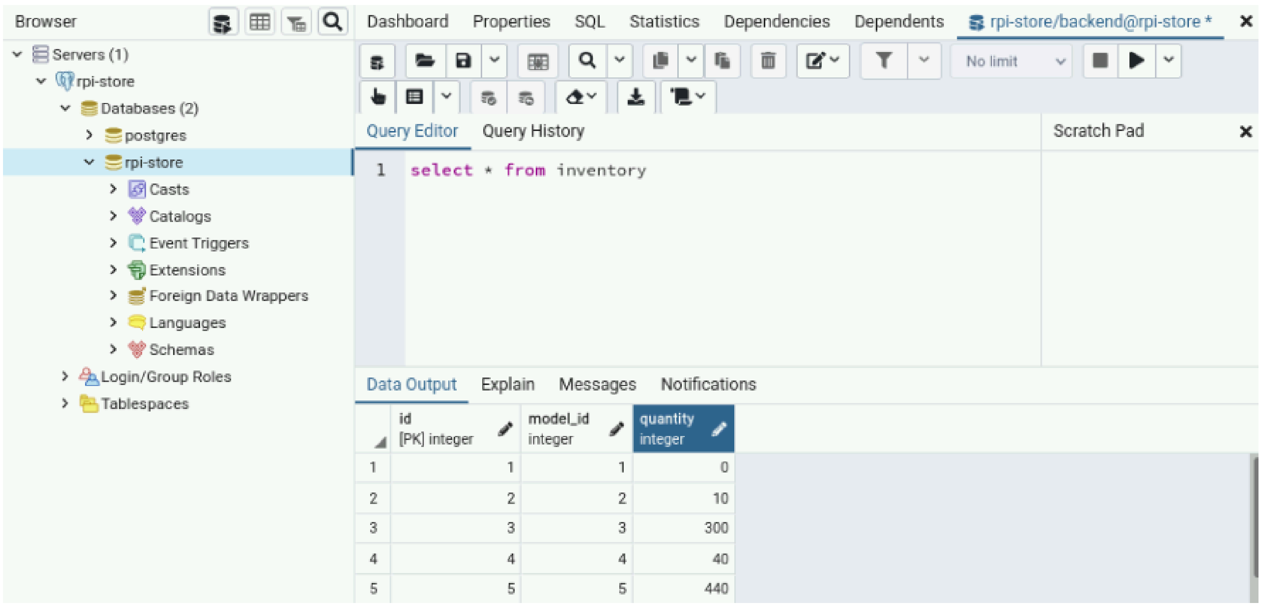
Press **F5** to execute the query and retrieve data from the `inventory` table.



```
select * from inventory
```

	id [PK] integer	model_id integer	quantity integer
1	1	1	0
2	2	2	20
3	3	3	300
4	4	4	40

Modify the data in the `inventory` table. Double-click the **20** value in the `quantity` column. Enter `10` as the value, press **Enter**, and then press **F6** to save the changes.



```
select * from inventory
```

	id [PK] integer	model_id integer	quantity integer
1	1	1	0
2	2	2	10
3	3	3	300
4	4	4	40
5	5	5	440

5. From your terminal, stop the development environment.

```
[student@workstation compose-environments]$ podman compose down
['podman', '--version', '']
using podman version: ...
** excluding:  set()
podman stop -t 10 compose_environments_postgresql
compose_environments_postgresql
exit code: 0
podman stop -t 10 compose_environments_pgadmin
compose_environments_pgadmin
exit code: 0
podman rm compose_environments_postgresql
738f...0506
exit code: 0
podman rm compose_environments_pgadmin
b584...670c
exit code: 0
```

Finish

On the `workstation` machine, change to the `student` user home directory and use the `lab` command to complete this exercise. This step is important to ensure that resources from previous exercises do not impact upcoming exercises.

```
[student@workstation ~]$ lab finish compose-environments
```