

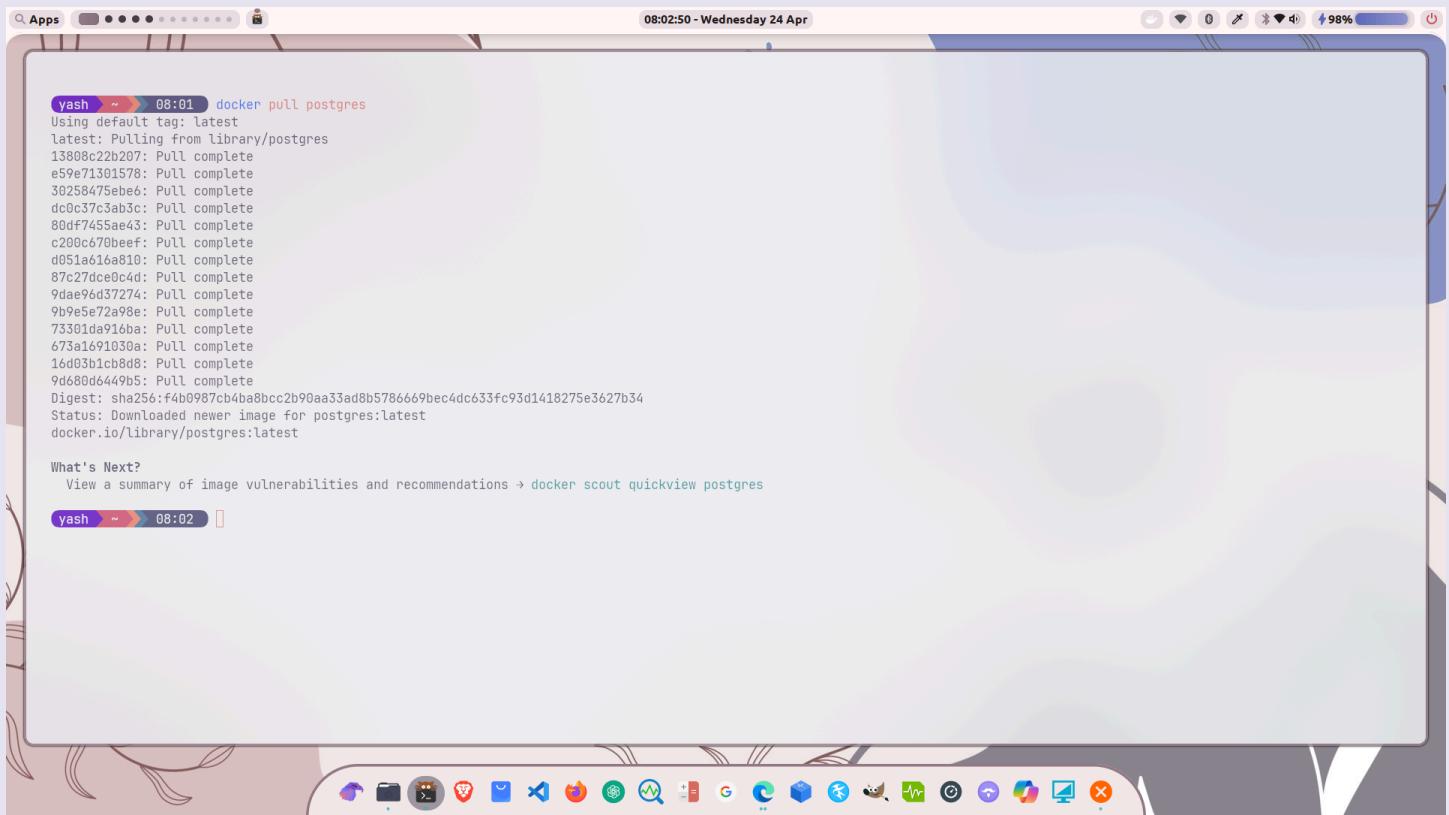
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

Aim : To understand how to containerize the existing Docker Images and Run the containers on Kubernetes Cluster.

Tasks and screenshots :

1. Consider an Image of Postgres. Pull the image from the Docker repository and run its container.

a) Pull the postgres image from dockerhub



The screenshot shows a Mac OS X desktop with a terminal window open. The terminal window has a light blue background and a dark blue header bar. The title bar says "yash ~ 08:01". The status bar at the top right shows "08:02:50 - Wednesday 24 Apr" and battery level "98%". The terminal window contains the following text:

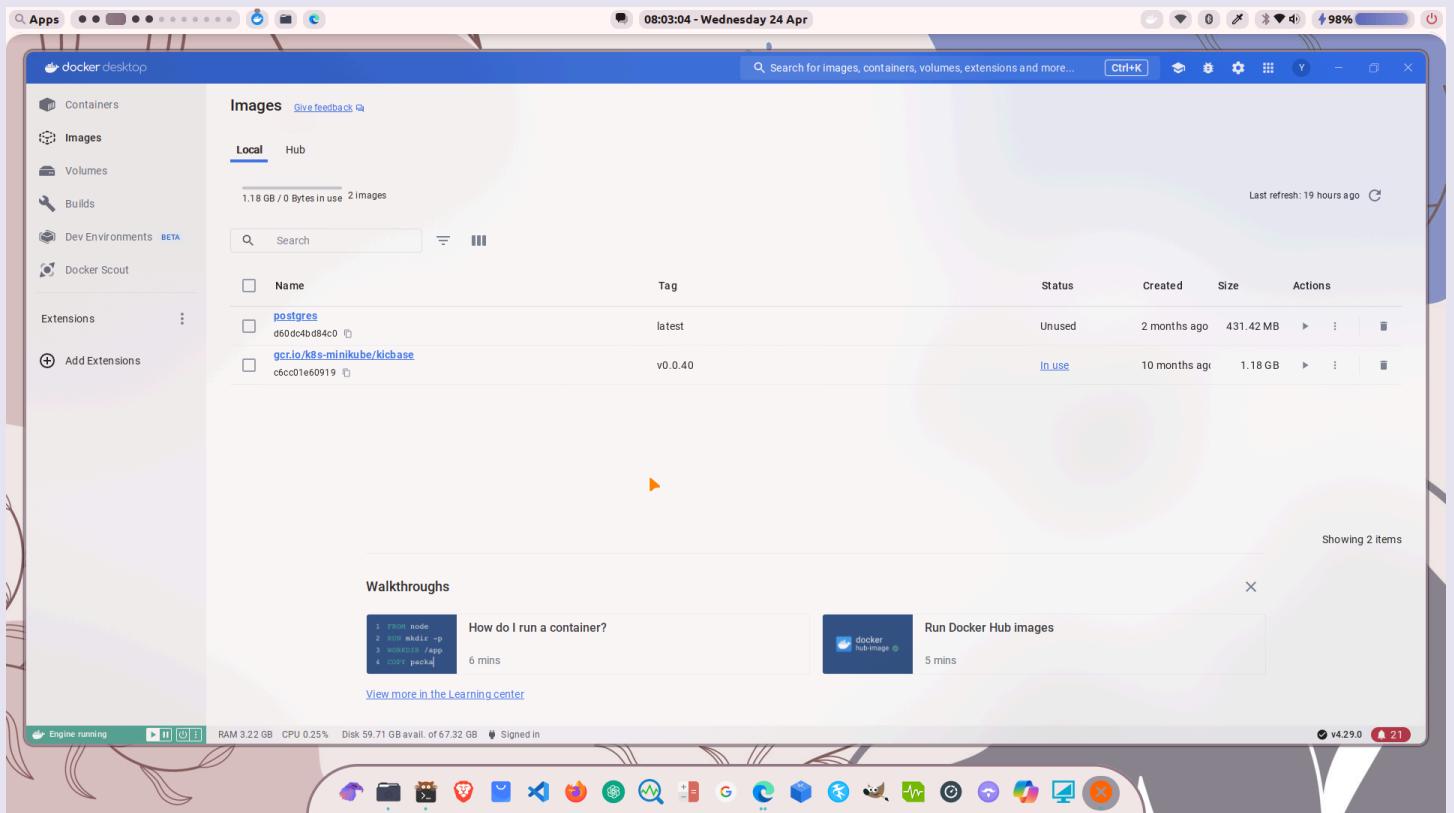
```
yash ~ 08:01 docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
13808c22b207: Pull complete
e59e71301578: Pull complete
30258475eb6: Pull complete
dc0c37c3ab3c: Pull complete
80df7455ae43: Pull complete
c200c670beef: Pull complete
d051a616a810: Pull complete
87c27dce0c4d: Pull complete
9dae96d37274: Pull complete
9b9e5e72a98e: Pull complete
73301da916ba: Pull complete
673a1691030a: Pull complete
16d03b1cb808: Pull complete
9d680d6449b5: Pull complete
Digest: sha256:f4b0987cb4ba8bcc2b90aa33ad8b5786669bec4dc633fc93d1418275e3627b34
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview postgres
```

The terminal window has a dark blue header bar with the title "yash ~ 08:01". The status bar at the bottom of the window shows "08:02". The desktop background is a light blue color with some decorative elements. The Dock at the bottom of the screen shows various application icons.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

b) Image is visible in docker desktop also and via `docker images` command also



Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

A screenshot of a macOS desktop environment. At the top, the Dock shows various application icons. The main focus is a terminal window titled 'yash ~' with a light blue background. The terminal displays the following command history:

```
yash ~ 08:01 docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
13808c22b207: Pull complete
e59e71301578: Pull complete
30258475eb6: Pull complete
dc0c37c3ab3c: Pull complete
80df7f455ae43: Pull complete
c200c670bee6: Pull complete
d051a616a810: Pull complete
87c27dce0c4d: Pull complete
9dae96d37274: Pull complete
9b9e5e72a98e: Pull complete
73301da916ba: Pull complete
673a1691030a: Pull complete
16d03b1cb8d8: Pull complete
9d680d6449b5: Pull complete
Digest: sha256:f4b0987cb4ba8bcc2b90aa33ad8b5786669bec4dc633fc93d1418275e3627b34
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview postgres
```

At the bottom of the terminal window, there is a small orange 'I' icon. The terminal prompt 'yash ~ 08:03 |' is visible at the bottom right.

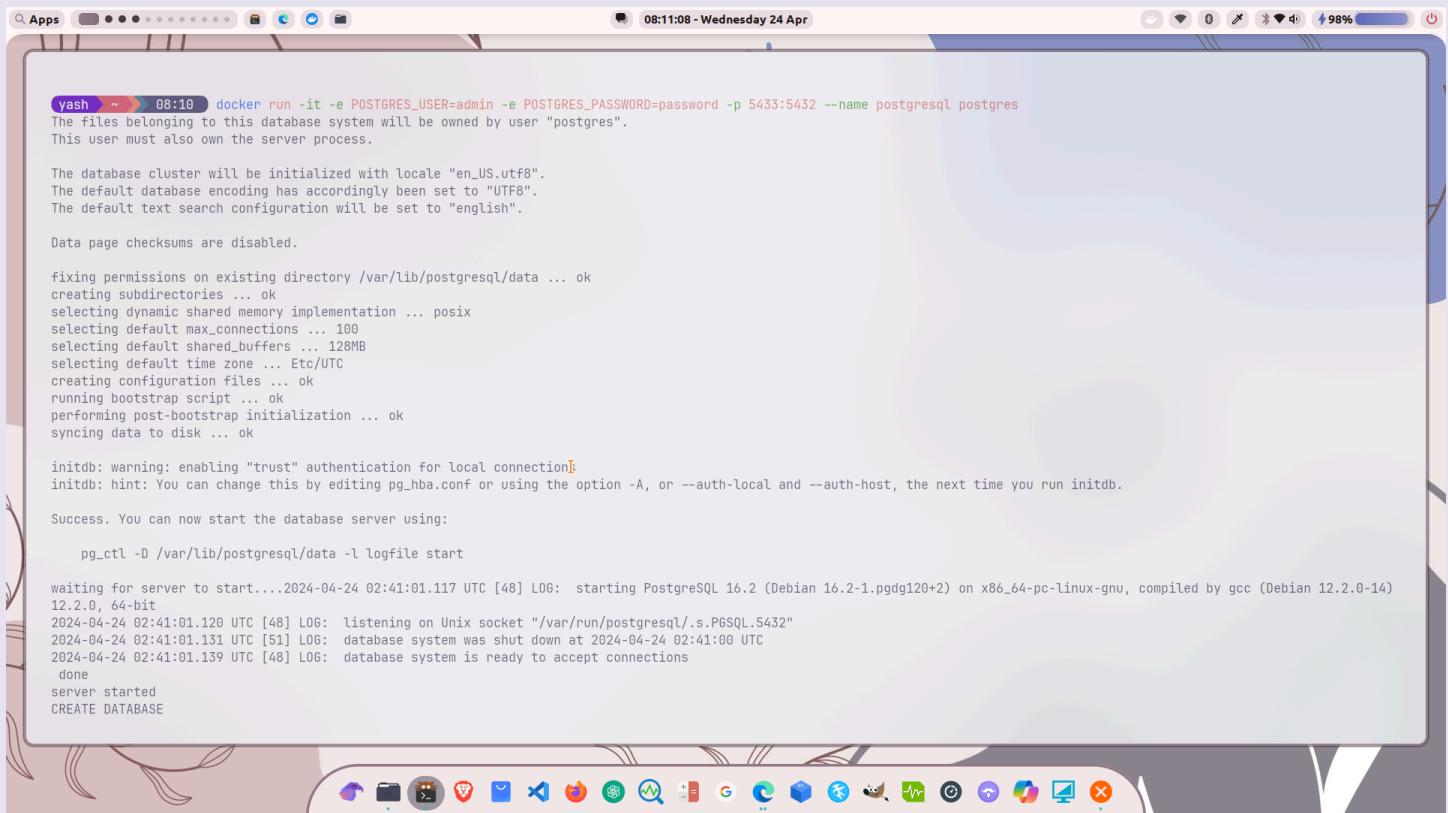
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

- c) Run the container on host port (here 5433) mapping with postgres port 5432 of container and specifying new user, password and running in interactive mode



A screenshot of a Linux desktop environment showing a terminal window. The terminal window has a light blue background and contains the following text:

```
yash ~ 08:10 docker run -it -e POSTGRES_USER=admin -e POSTGRES_PASSWORD=password -p 5433:5432 --name postgresql postgres
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.utf8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/data ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... Etc/UTC
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connection!
initdb: hint: You can change this by editing pg_hba.conf or using the option -A, or --auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

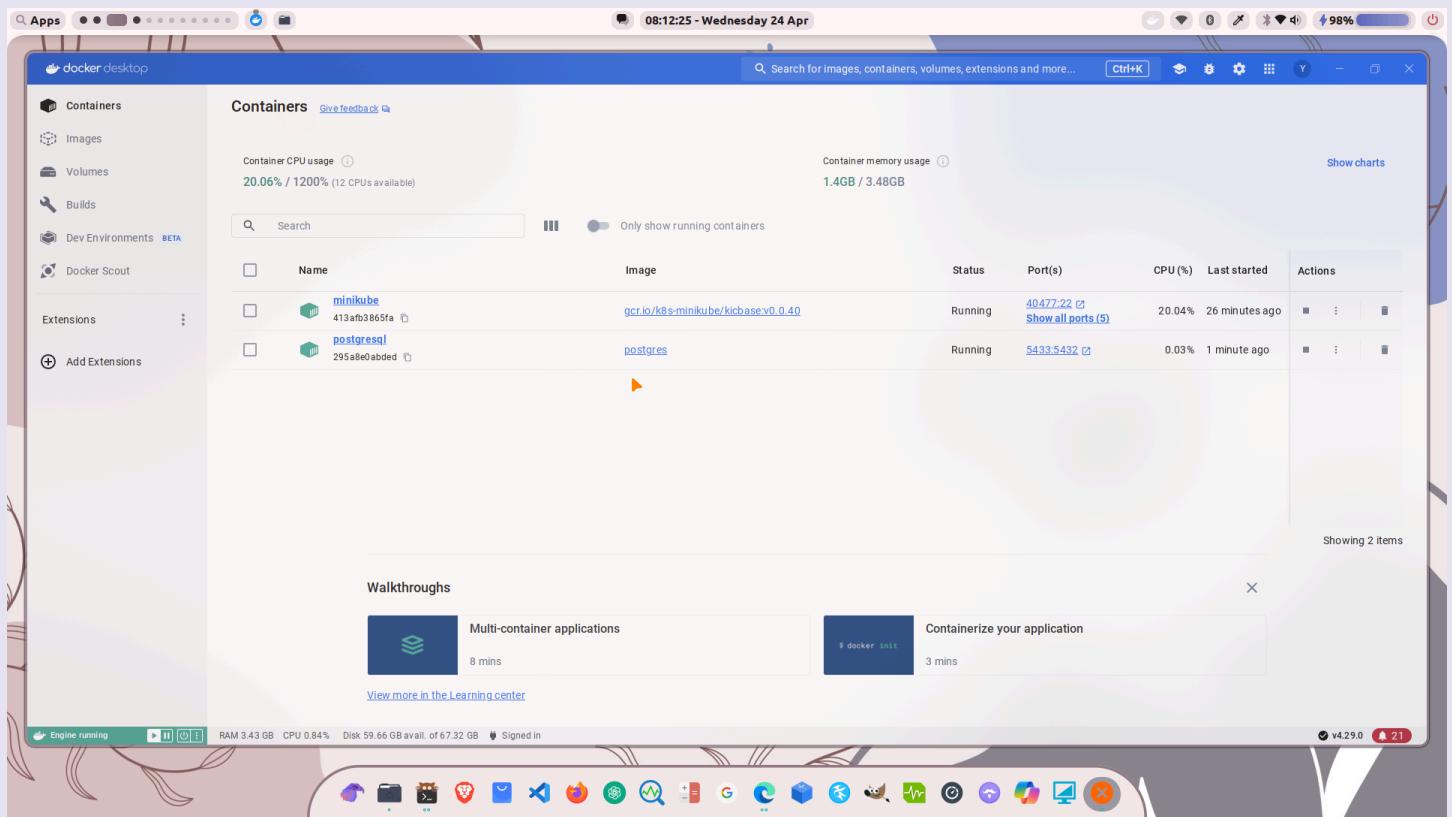
  pg_ctl -D /var/lib/postgresql/data -l logfile start

waiting for server to start...2024-04-24 02:41:01.117 UTC [48] LOG:  starting PostgreSQL 16.2 (Debian 16.2-1.pgdg120+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14)
12.2.0, 64-bit
2024-04-24 02:41:01.120 UTC [48] LOG:  listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
2024-04-24 02:41:01.131 UTC [51] LOG:  database system was shut down at 2024-04-24 02:41:00 UTC
2024-04-24 02:41:01.139 UTC [48] LOG:  database system is ready to accept connections
done
server started
CREATE DATABASE
```

The terminal window is titled "08:11:08 - Wednesday 24 Apr". The desktop environment includes a dock with various application icons at the bottom.

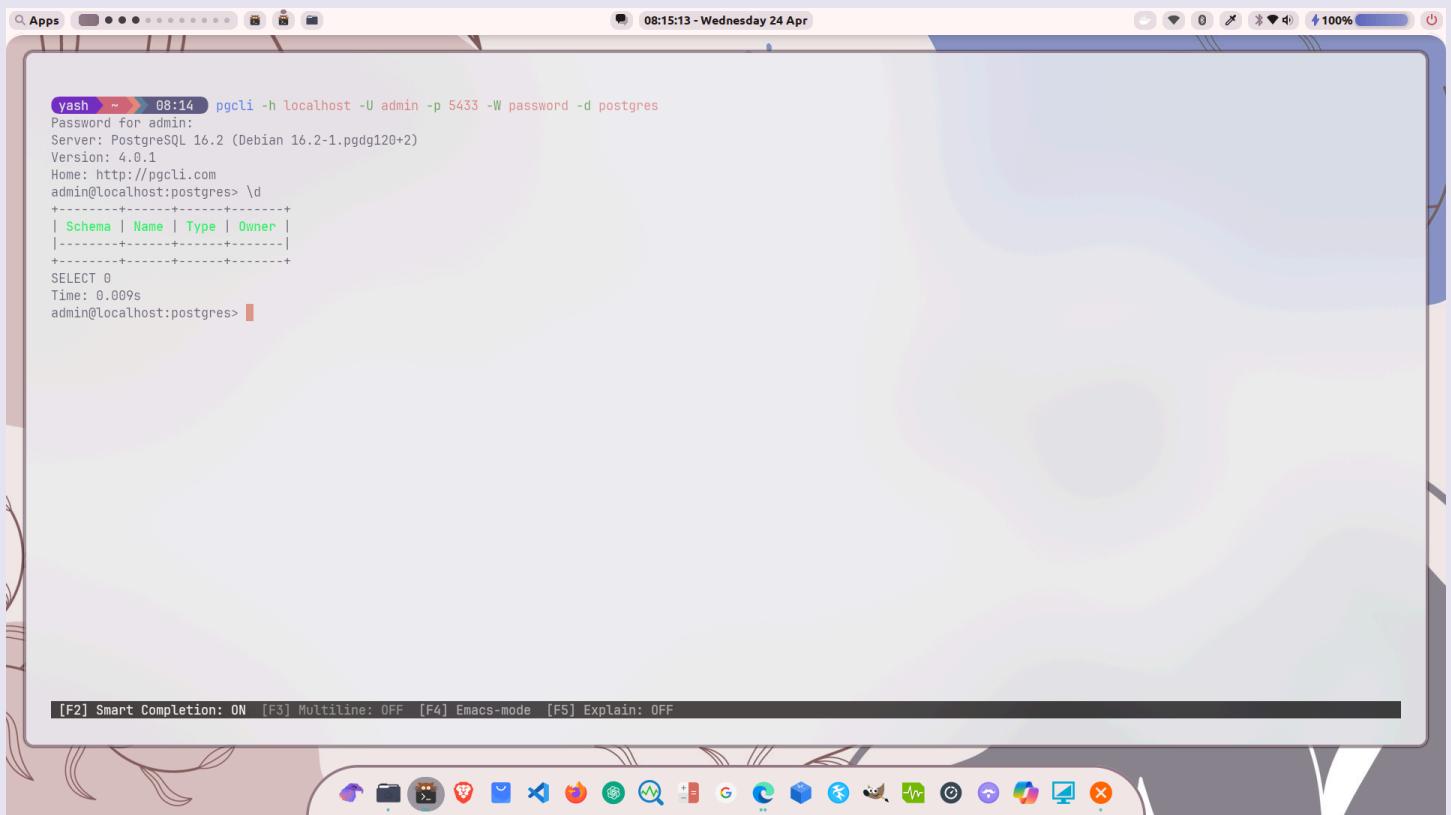
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

d) The running container is visible on docker desktop also



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

e) Try connect with psql on local machine



A screenshot of a Linux desktop environment. The terminal window shows the command `pgcli -h localhost -U admin -p 5433 -W password -d postgres` being run. The output indicates a successful connection to PostgreSQL 16.2 on port 5433. The user is then prompted to enter a password for the 'admin' user. The terminal also displays the schema list command (`\d`) and its results, showing one schema named 'public'. The bottom status bar of the terminal window shows keyboard shortcuts for Smart Completion (F2), Multiline (F3), Emacs-mode (F4), and Explain (F5). The desktop background features a light blue and white floral pattern. The system tray at the bottom of the screen shows various icons for system monitoring and connectivity.

```
yash ~ 08:14 pgcli -h localhost -U admin -p 5433 -W password -d postgres
Password for admin:
Server: PostgreSQL 16.2 (Debian 16.2-1.pgdg120+2)
Version: 4.0.1
Home: http://pgcli.com
admin@localhost:postgres> \d
Schema | Name | Type | Owner |
-----+-----+-----+
public
SELECT 0
Time: 0.009s
admin@localhost:postgres>
```

(Here, pgcli is used instead of psql, which has same syntax and options, but includes features like autocomplete, syntax highlighting and much more)

Below are provided options, works also for [psql](#)

Specify details	Option
username	-U
password	-W
host	-h
database	-d
port	-p

Name - Yash Lakhtariya

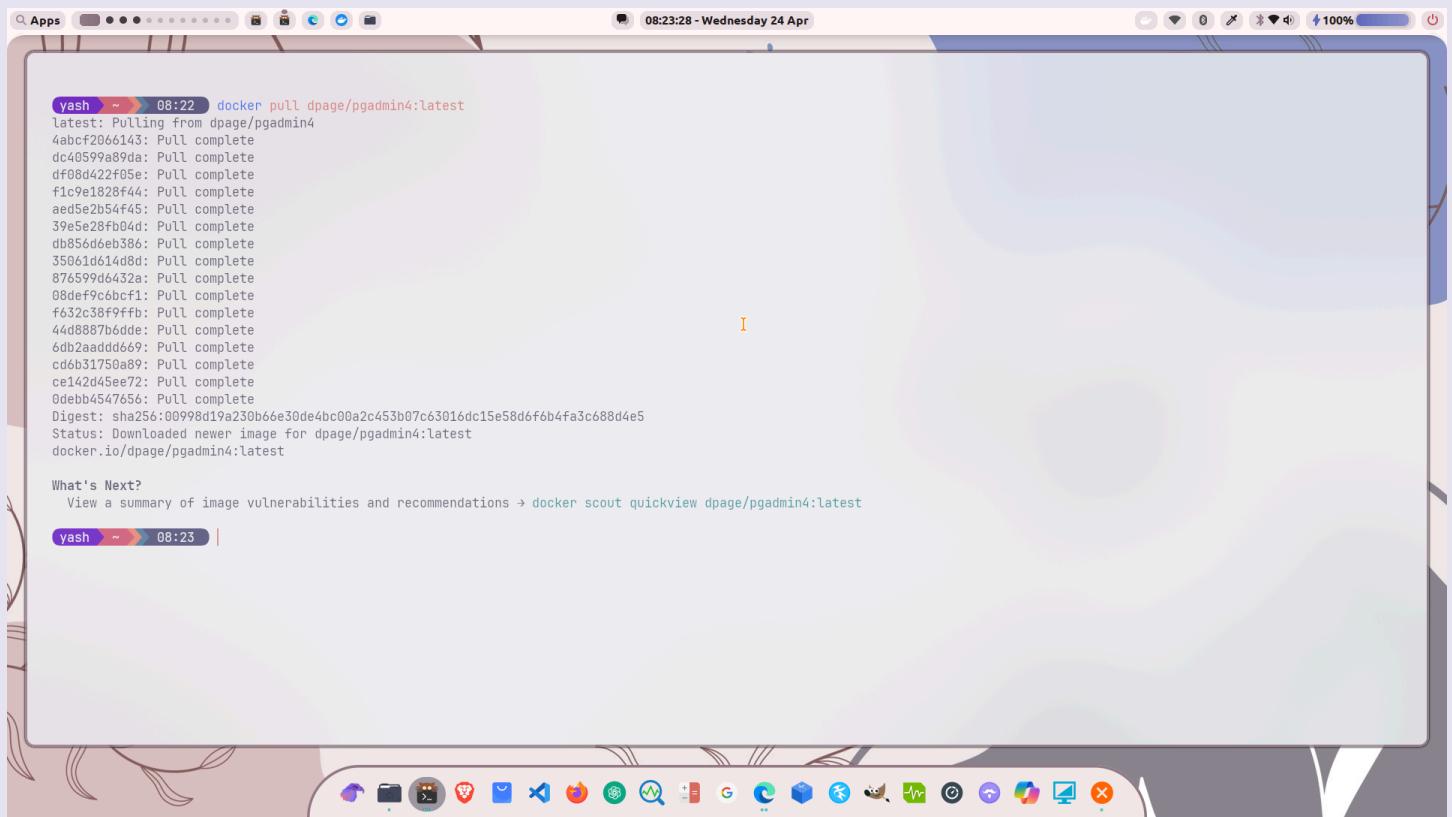
Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

2. Pull the image of the Database client (pg admin) in this case, and run its container.

- Pull pgadmin image from docker hub



A screenshot of a Mac OS X desktop environment. In the center is a terminal window titled '08:23:28 - Wednesday 24 Apr'. The terminal shows the command 'docker pull dpage/pgadmin4:latest' being run at 08:22. The output of the command is displayed, showing the progress of pulling the image from the Docker Hub. The progress bar is nearly complete, indicating that the pull operation has finished. The status message at the end of the output indicates that a newer image is available on Docker Hub. The desktop background features a light blue and white abstract design. The Dock at the bottom of the screen contains various application icons, including Finder, Mail, Safari, and others. The system tray at the top right shows battery level (100%), signal strength, and other system status indicators.

```
yash ~ 08:22 docker pull dpage/pgadmin4:latest
latest: Pulling from dpage/pgadmin4
4abcf2066143: Pull complete
dc40599a89da: Pull complete
df08d42f05e: Pull complete
f1c9e1828f44: Pull complete
aed5e2b54f45: Pull complete
39e5e28fb04d: Pull complete
db856d6eb386: Pull complete
35061d614d8d: Pull complete
876599d6432a: Pull complete
08def9c6bcfc1: Pull complete
f632c38f9ffb: Pull complete
44d8887b6dde: Pull complete
6db2aaddde69: Pull complete
cd6b31750a89: Pull complete
ce142d45ee72: Pull complete
0debb4547656: Pull complete
Digest: sha256:00998d19a230b66e30de4bc00a2c453b07c63016dc15e58d6f6b4fa3c688d4e5
Status: Downloaded newer image for dpage/pgadmin4:latest
docker.io/dpage/pgadmin4:latest

What's Next?
View a summary of image vulnerabilities and recommendations → docker scout quickview dpage/pgadmin4:latest
```

yash ~ 08:23 |

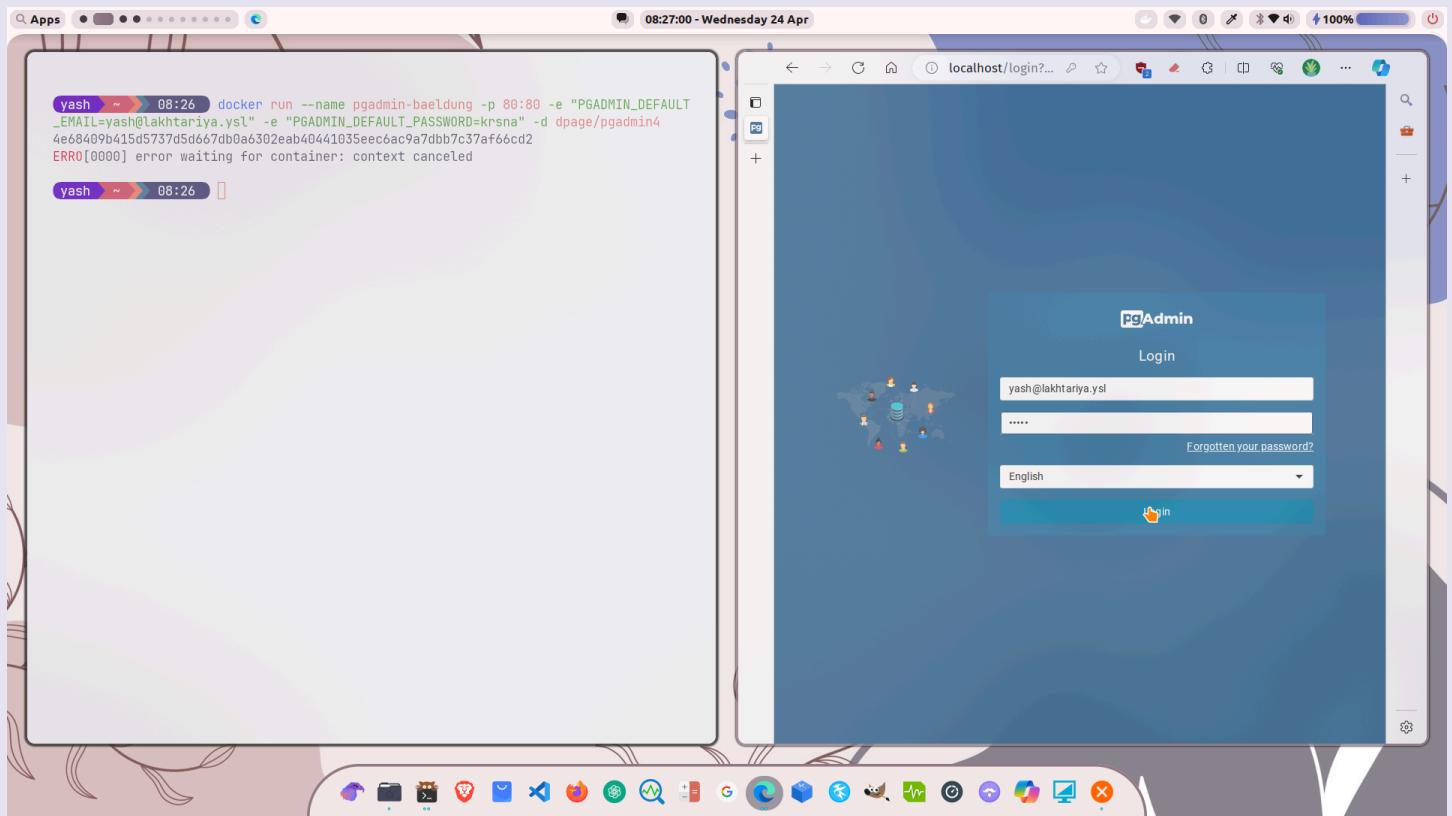
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

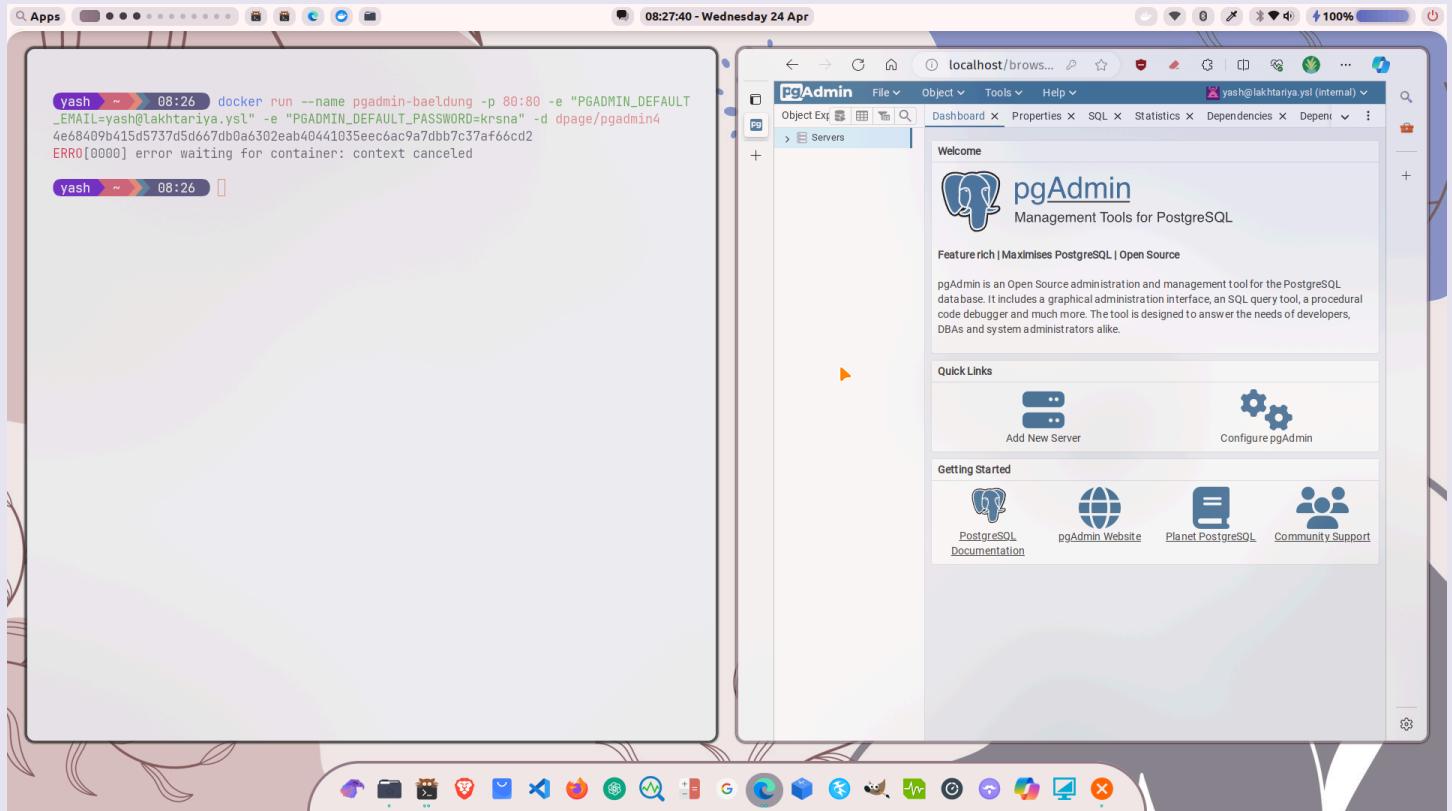
EADC Practical 20

- b) Run container of it in detached mode (option -d), and specify pgadmin default email and password to use initially



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

c) The container is successfully running



Name - Yash Lakhtariya

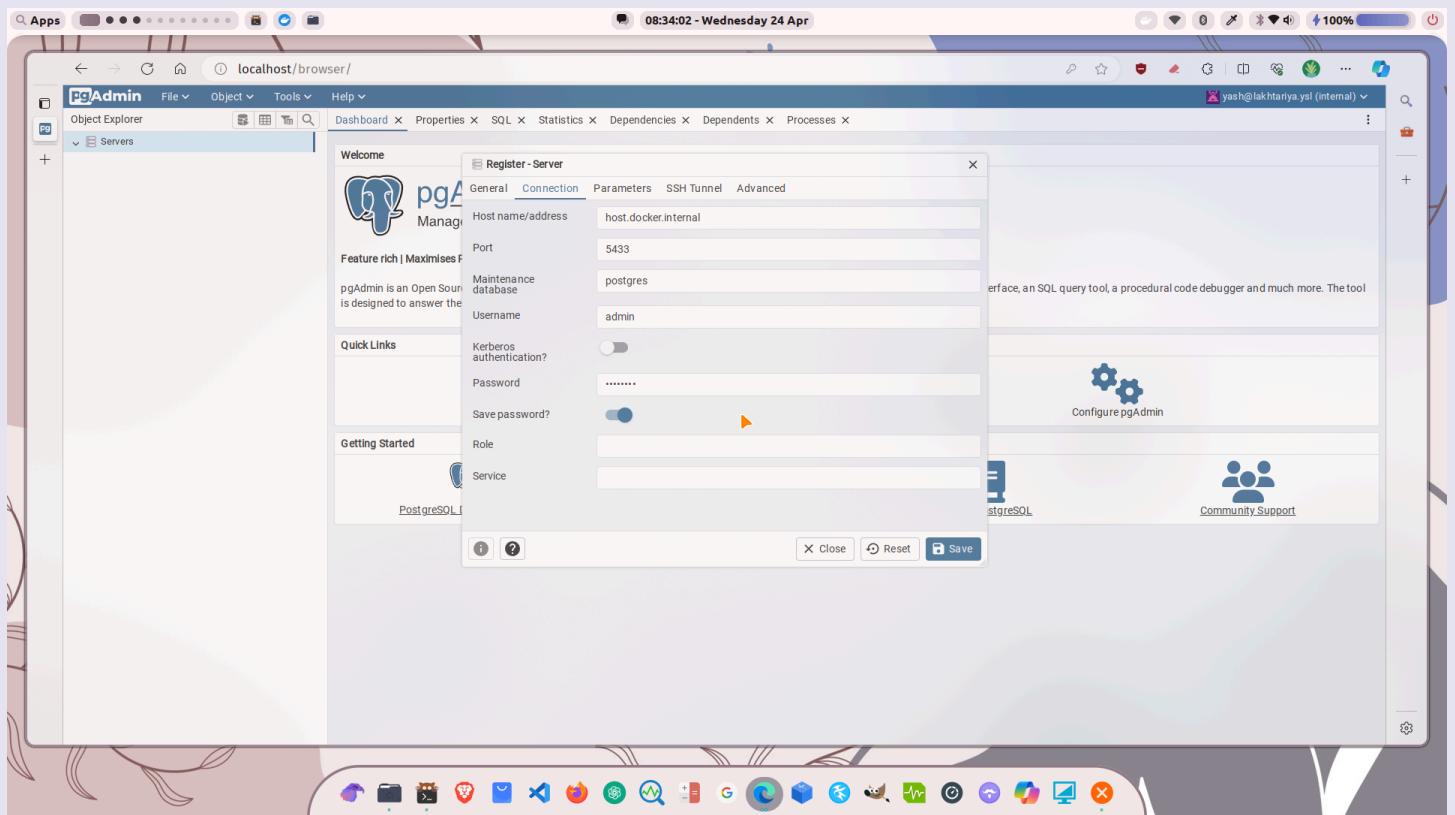
Enrollment number - 21162101012

Branch - CBA Batch - 61

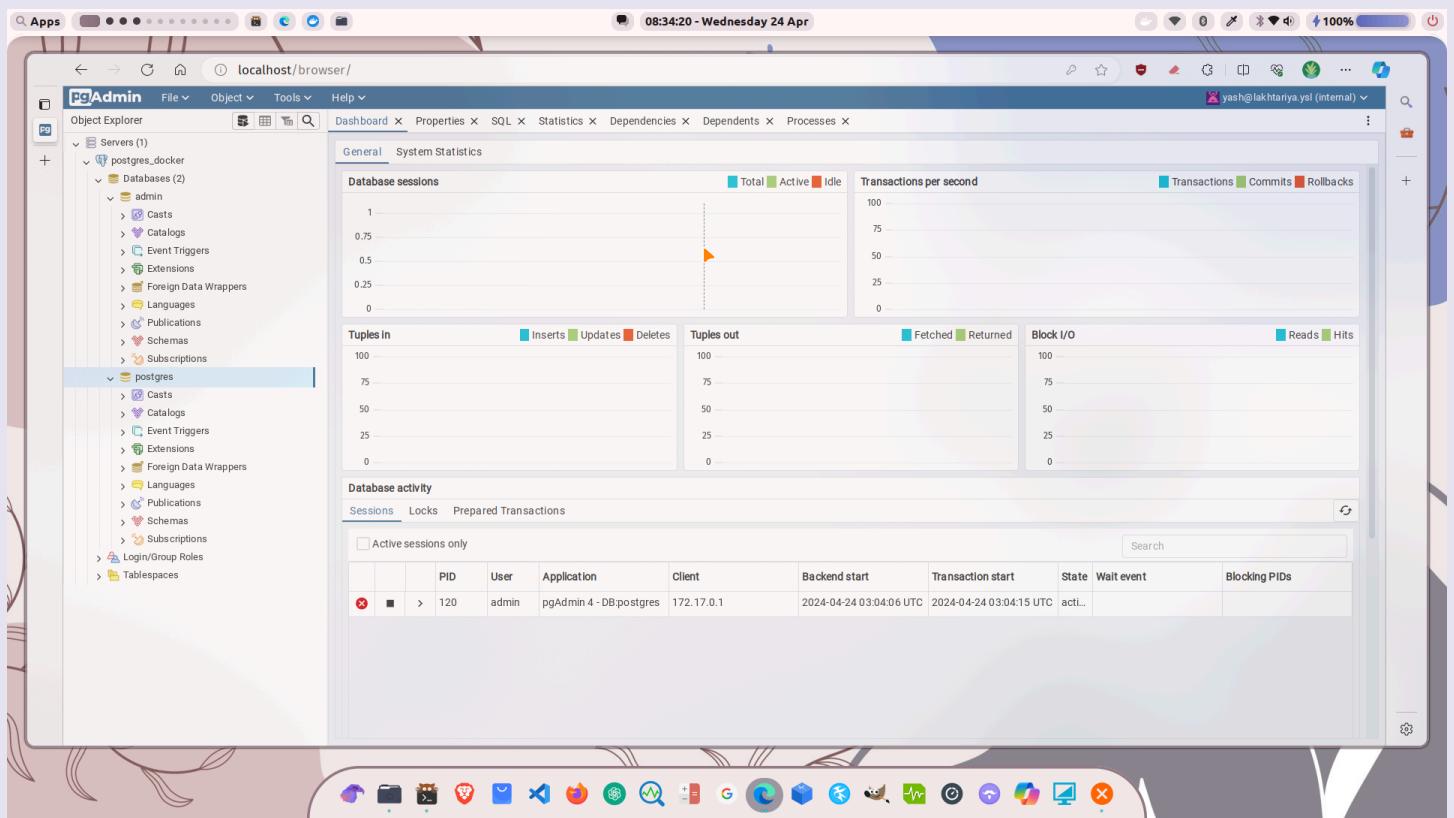
EADC Practical 20

3. Make the Postgres database and PG admin containers communicate with each other.

- Register server on pgadmin with the given details of container postgres



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

- b) Create database using psql in postgres container and check after Refresh on pgadmin container, if successful connection is there, it will show in the list, (here ysl database)

The screenshot displays a desktop environment with two windows open. On the left is the pgAdmin interface, which shows a tree view of servers and databases. A database named 'ysl' is selected. On the right is a terminal window showing the creation of a database named 'ysl' using the pgcli command.

pgAdmin Interface:

- Servers (1)
 - postres_docker
 - Databases (3)
 - admin
 - postgres
 - ysl
 - Catalogs
 - Event Triggers
 - Extensions
 - Foreign Data W
 - Languages
 - Publications
 - Schemas
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

Terminal Window:

```
yash ~ 08:34 | pgcli -h localhost -U admin -p 5433 -W password -d postgres
Password for admin:
Server: PostgreSQL 16.2 (Debian 16.2-1.pgdg120+2)
Version: 4.0.1
Home: http://pgcli.com
admin@localhost:postgres> create database ysl;
CREATE DATABASE
Time: 0.105s
admin@localhost:postgres> show databases;
unrecognized configuration parameter "databases"
Time: 0.002s
admin@localhost:postgres>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

c) Perform some operations like creating table, inserting data, etc.

The screenshot shows a Mac desktop with two windows open. On the left is the pgAdmin 4 interface, connected to a 'postgres_docker' server. The 'Databases' section shows three databases: admin, postgres, and ysl. The 'Dashboard' tab is active, displaying several performance metrics: 'Database sessions' (1), 'Transactions per second' (approx. 1.5), 'Tuples in' (approx. 100), 'Tuples out' (approx. 100), and 'Block I/O' (approx. 1,000). On the right is a terminal window titled 'yash ~ 08:34'. It shows the creation of a new database named 'ysl' using pgcli:

```
yash ~ 08:34 | pgcli -h localhost -U admin -p 5433 -W password -d postgres
Password for admin:
Server: PostgreSQL 16.2 (Debian 16.2-1.pgdg120+2)
Version: 4.0.1
Home: http://pgcli.com
admin@localhost:postgres> create database ysl;
CREATE DATABASE
Time: 0.105s
admin@localhost:postgres> show databases;
unrecognized configuration parameter "databases"
Time: 0.002s
admin@localhost:postgres> show database
unrecognized configuration parameter "database"
Time: 0.002s
admin@localhost:postgres> \list
+-----+-----+-----+-----+-----+
| Name | Owner | Encoding | Collate | Ctype | Access privileges |
+-----+-----+-----+-----+-----+
| admin | admin | UTF8 | en_US.utf8 | en_US.utf8 | <null>
| postgres | admin | UTF8 | en_US.utf8 | en_US.utf8 | <null>
| template0 | admin | UTF8 | en_US.utf8 | en_US.utf8 | =c/admin
| template1 | admin | UTF8 | en_US.utf8 | en_US.utf8 | =c/admin
| ysl | admin | UTF8 | en_US.utf8 | en_US.utf8 | admin=CTc/admin
+-----+-----+-----+-----+-----+
SELECT 5
Time: 0.009s
admin@localhost:postgres>
```

The terminal also displays the prompt '[F2] Smart Completion: ON [F3] Multiline: OFF [F4] Emacs-mode [F5] Explain: OFF'.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer tree view shows the database structure: 'Servers' > 'postgres_docker' > 'Databases' > 'ysl' > 'Schemas' > 'public' > 'Tables' > 'users'. The 'users' table is selected. The main pane displays the 'Properties' tab for the 'users' table, with the 'General' section active. The table has the following properties:

- Name: users
- OID: 16398
- Owner: admin
- Tablespace: pg_default
- Partitioned table?: Off
- System table?: Off
- Comment: (empty)

The 'Advanced' section is collapsed. The status bar at the bottom indicates: [F2] Smart Completion: ON [F3] Multiline: OFF [F4] Emacs-mode [F5] Explain: OFF.

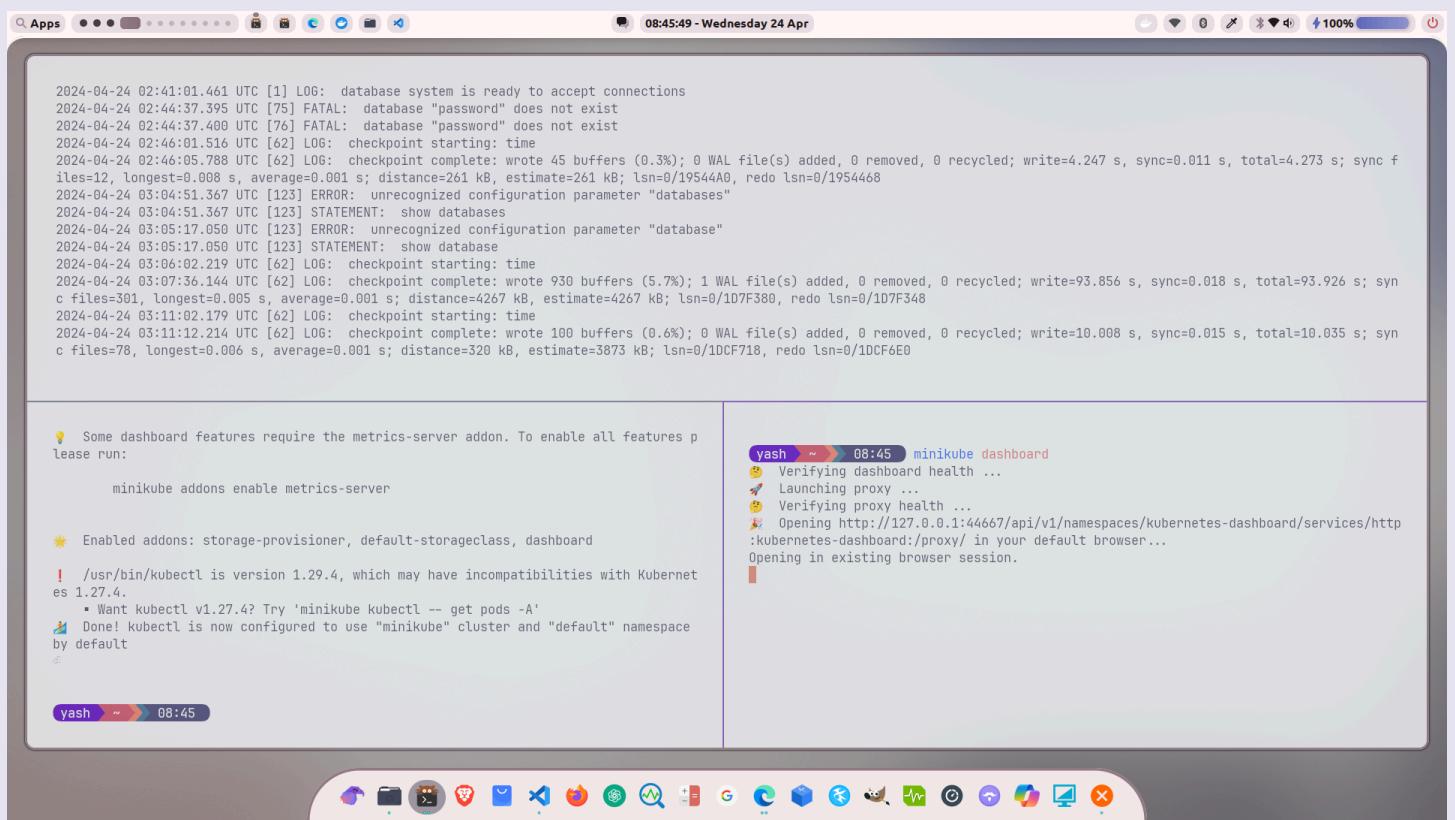
On the right, a terminal window shows the following PostgreSQL session:

```
08:39:37 - Wednesday 24 Apr
SELECT 2
Time: 0.005s
admin@localhost:postgres> \c ysl;
You are now connected to database "ysl" as user "admin"
Time: 0.011s
admin@localhost:ysl> CREATE TABLE users (id SERIAL PRIMARY KEY, name
    | VARCHAR(30), email VARCHAR(30));
CREATE TABLE
Time: 0.014s
admin@localhost:ysl> INSERT INTO users (name, email) VALUES ('Jerry',
    | 'jerry@example.com'), ('George', 'george@example.com');
INSERT 0 2
Time: 0.007s
admin@localhost:ysl> \d
+-----+-----+-----+
| Schema | Name      | Type   | Owner |
+-----+-----+-----+
| public | users     | table  | admin  |
| public | users_id_seq | sequence | admin  |
+-----+-----+-----+
SELECT 2
Time: 0.005s
admin@localhost:ysl> select * from users;
+-----+-----+
| id | name      | email            |
+-----+-----+
| 1  | Jerry     | jerry@example.com |
| 2  | George    | george@example.com |
+-----+-----+
SELECT 2
Time: 0.004s
admin@localhost:ysl>
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

4. Run Postgres on Kubernetes and learn to Externalize config using ConfigMaps in Kubernetes.

a) Start minikube and its dashboard



08:45:49 - Wednesday 24 Apr

```
2024-04-24 02:41:01.461 UTC [1] LOG:  database system is ready to accept connections
2024-04-24 02:44:37.395 UTC [75] FATAL:  database "password" does not exist
2024-04-24 02:44:37.400 UTC [76] FATAL:  database "password" does not exist
2024-04-24 02:46:01.516 UTC [62] LOG:  checkpoint starting: time
2024-04-24 02:46:05.788 UTC [62] LOG:  checkpoint complete: wrote 45 buffers (0.3%); 0 WAL file(s) added, 0 removed, 0 recycled; write=4.247 s, sync=0.011 s, total=4.273 s; sync files=12, longest=0.008 s, average=0.001 s; distance=261 kB, estimate=261 kB; lsn=0/19544A0, redo lsn=0/1954468
2024-04-24 03:04:51.367 UTC [123] ERROR:  unrecognized configuration parameter "databases"
2024-04-24 03:04:51.367 UTC [123] STATEMENT: show databases
2024-04-24 03:05:17.050 UTC [123] ERROR:  unrecognized configuration parameter "database"
2024-04-24 03:05:17.050 UTC [123] STATEMENT: show database
2024-04-24 03:06:02.219 UTC [62] LOG:  checkpoint starting: time
2024-04-24 03:07:36.144 UTC [62] LOG:  checkpoint complete: wrote 930 buffers (5.7%); 1 WAL file(s) added, 0 removed, 0 recycled; write=93.856 s, sync=0.018 s, total=93.926 s; sync files=301, longest=0.005 s, average=0.001 s; distance=4267 kB, estimate=4267 kB; lsn=0/1D7F380, redo lsn=0/1D7F346
2024-04-24 03:11:02.179 UTC [62] LOG:  checkpoint starting: time
2024-04-24 03:11:12.214 UTC [62] LOG:  checkpoint complete: wrote 100 buffers (0.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=10.008 s, sync=0.015 s, total=10.035 s; sync files=78, longest=0.006 s, average=0.001 s; distance=320 kB, estimate=3873 kB; lsn=0/1DCF718, redo lsn=0/1DCF6E0
```

Some dashboard features require the metrics-server addon. To enable all features please run:

```
minikube addons enable metrics-server
```

Enabled addons: storage-provisioner, default-storageclass, dashboard

/usr/bin/kubectl is version 1.29.4, which may have incompatibilities with Kubernetes 1.27.4.

- Want kubectl v1.27.4? Try 'minikube kubectl -- get pods -A'

Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

yash ~ 08:45 minikube dashboard

Verifying dashboard health ...
Launching proxy ...
Verifying proxy health ...
Opening http://127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
Opening in existing browser session.

yash ~ 08:45

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

b) Open dashboard and add config map file to it

The screenshot shows a web browser window displaying the Kubernetes Dashboard at the URL `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/configmap?namespace=default`. The title bar indicates the time is 08:47:40 on Wednesday, April 24. The dashboard has a blue header with the title "Config And Storage > Config Maps". On the left, there is a sidebar with navigation links for Workloads, Service, Config and Storage, and Cluster. Under Config and Storage, the "Config Maps" link is highlighted. The main content area shows a table titled "Config Maps" with one entry: "kube-root-ca.crt". The table includes columns for Name, Labels, and Created. The "Created" column shows "59 minutes ago". A "Create new resource" button is located in the top right corner of the main content area.

Name	Labels	Created
kube-root-ca.crt	-	59 minutes ago

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

The screenshot shows a web browser window for the Kubernetes Dashboard. The URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/create?namespace=default`. The title bar indicates the time is 08:48:02 on Wednesday, April 24. The dashboard has a sidebar on the left with categories like Workloads, Service, Config and Storage, and Cluster. The main area is titled "Create" and shows a "ConfigMaps" section. A modal dialog is open, prompting the user to "Enter YAML or JSON content specifying the resources to create to the currently selected namespace". The content area contains the following YAML code:

```
1 apiVersion: v1
2 kind: ConfigMap
3 metadata:
4   name: postgres-config
5   labels:
6     app: postgres
7 data:
8   POSTGRES_DB: postgresdb
9   POSTGRES_USER: admin
10  POSTGRES_PASSWORD: test123
```

At the bottom of the modal, there are "Upload" and "Cancel" buttons. The "Upload" button is highlighted with a blue border.

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

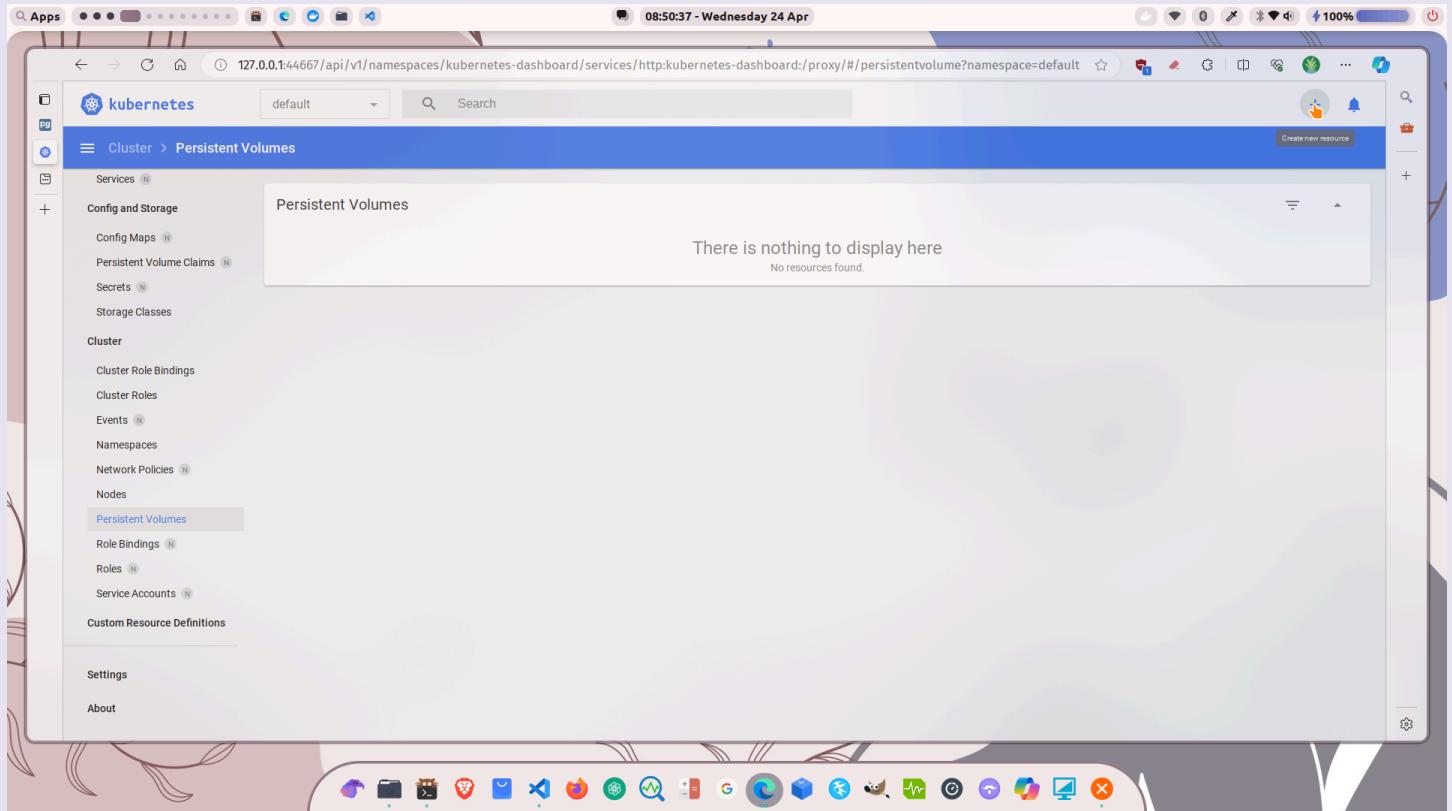
EADC Practical 20

The screenshot shows the Kubernetes Dashboard interface running locally at `127.0.0.1:44667`. The main title bar indicates the URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/configmap?namespace=default`. The top navigation bar shows the date and time as `08:48:47 - Wednesday 24 Apr`. The dashboard has a sidebar on the left with categories: Workloads, Service, Config and Storage, and Cluster. Under Config and Storage, the `Config Maps` tab is selected. The main content area displays a table titled "Config Maps" with two entries:

Name	Labels	Created
postgres-config	app:postgres	42 seconds ago
kube-root-ca.crt	-	an hour ago

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

c) Add persistent volume



The screenshot shows a web browser window displaying the Kubernetes Dashboard. The URL in the address bar is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/persistentvolume?namespace=default`. The title bar indicates the time is 08:50:37 on Wednesday, April 24. The dashboard has a blue header with the text "Cluster > Persistent Volumes". On the left, there is a sidebar with various navigation options: Services, Config and Storage (Config Maps, Persistent Volume Claims, Secrets), Storage Classes, Cluster (Cluster Role Bindings, Cluster Roles, Events, Namespaces, Network Policies, Nodes), Persistent Volumes (selected), Role Bindings, Roles, Service Accounts, Custom Resource Definitions, Settings, and About. The main content area is titled "Persistent Volumes" and contains the message "There is nothing to display here" and "No resources found". A "Create new resource" button is located in the top right corner of this section. The bottom of the screen shows a dock with various application icons.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

The screenshot shows the Kubernetes Dashboard interface on a Mac OS X desktop. The URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/create?namespace=default`. The title bar indicates the time is 08:50:47 on Wednesday, April 24. The main window is titled "Create" and shows a "Config and Storage" section with "Persistent Volumes" selected. A YAML editor displays the following code:

```
1 kind: PersistentVolume
2 apiVersion: v1
3 metadata:
4   name: postgres-pv-volume
5   labels:
6     type: local
7     app: postgres
8   spec:
9     storageClassName: manual
10    capacity:
11      storage: 5Gi
12    accessModes:
13      - ReadWriteMany
14    hostPath:
15      path: "/mnt/data"
```

At the bottom of the editor are "Upload" and "Cancel" buttons. The status bar at the bottom of the screen shows various application icons.

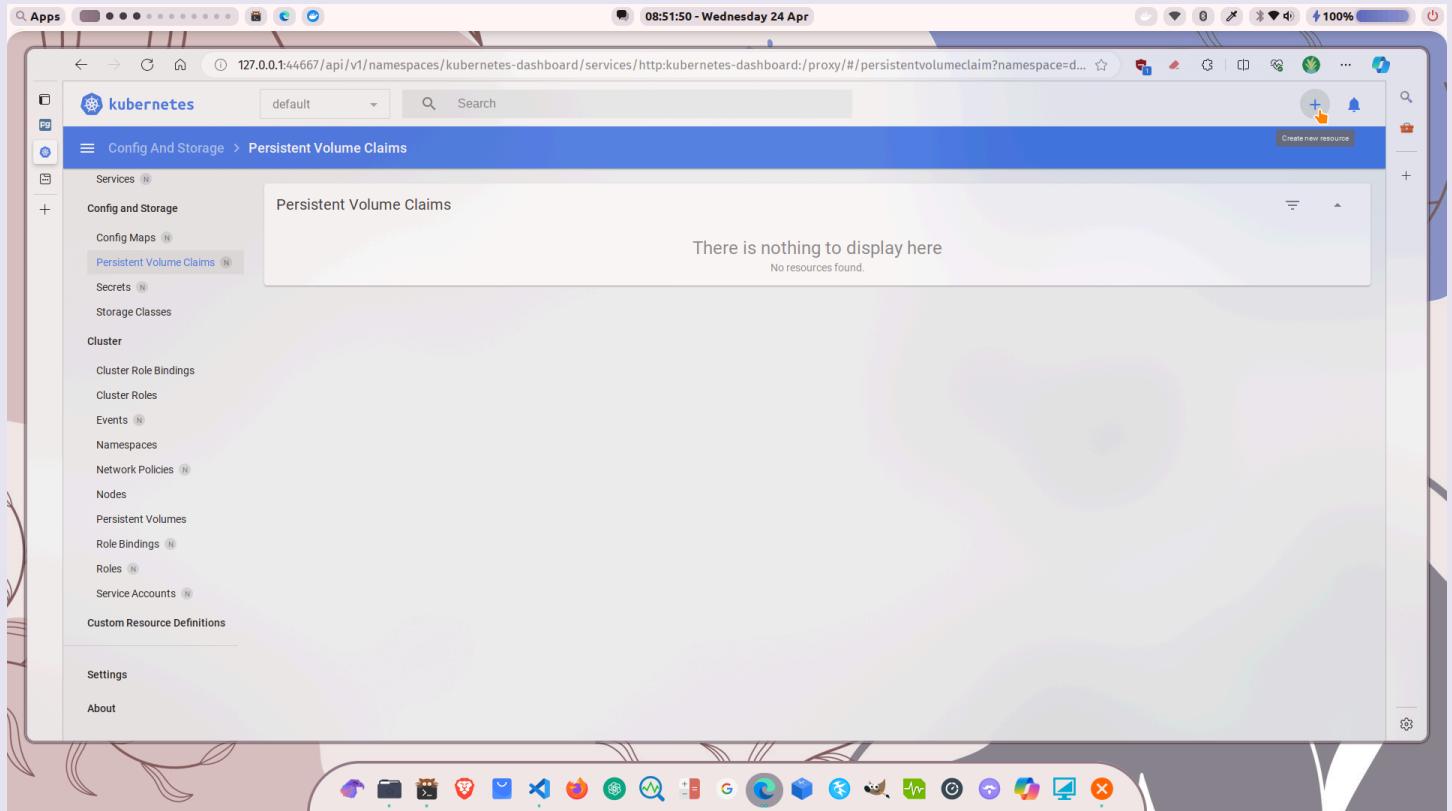
The screenshot shows the Kubernetes Dashboard interface on a Mac OS X desktop. The URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/persistentvolume?namespace=default`. The title bar indicates the time is 08:50:57 on Wednesday, April 24. The main window is titled "Cluster > Persistent Volumes" and shows a "Persistent Volumes" table. The table has the following columns: Name, Capacity, Access Modes, Reclaim Policy, Status, Claim, Storage Class, Reason, and Created. One entry is listed:

Name	Capacity	Access Modes	Reclaim Policy	Status	Claim	Storage Class	Reason	Created
postgres-pv-volume	storage: 5Gi	ReadWriteMany	Retain	Available	-	manual	-	6 seconds ago

The status bar at the bottom of the screen shows various application icons.

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

d) Similarly, add volume claim



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

The screenshot shows the Kubernetes dashboard interface on a Mac OS X desktop. The URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/create?namespace=default`. The title bar indicates the time is 08:52:01 on Wednesday, April 24. The main window is titled "Create" and has a sub-header "Config and Storage". It displays a YAML configuration for a PersistentVolumeClaim:

```
1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: postgres-pv-claim
5   labels:
6     app: postgres
7 spec:
8   storageClassName: manual
9   accessModes:
10    - ReadWriteMany
11   resources:
12     requests:
13       storage: 5Gi
```

At the bottom of the form are "Upload" and "Cancel" buttons. The "Upload" button is highlighted with a cursor.

The screenshot shows the Kubernetes dashboard interface on a Mac OS X desktop. The URL is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/persistentvolumeclaim?namespace=d...`. The title bar indicates the time is 08:52:09 on Wednesday, April 24. The main window is titled "Config And Storage > Persistent Volume Claims". It displays a table of Persistent Volume Claims:

Name	Labels	Status	Volume	Capacity	Access Modes	Storage Class	Created
postgres-pv-claim	app: postgres	Bound	postgres-pv-volume	5Gi	ReadWriteMany	manual	4 seconds ago

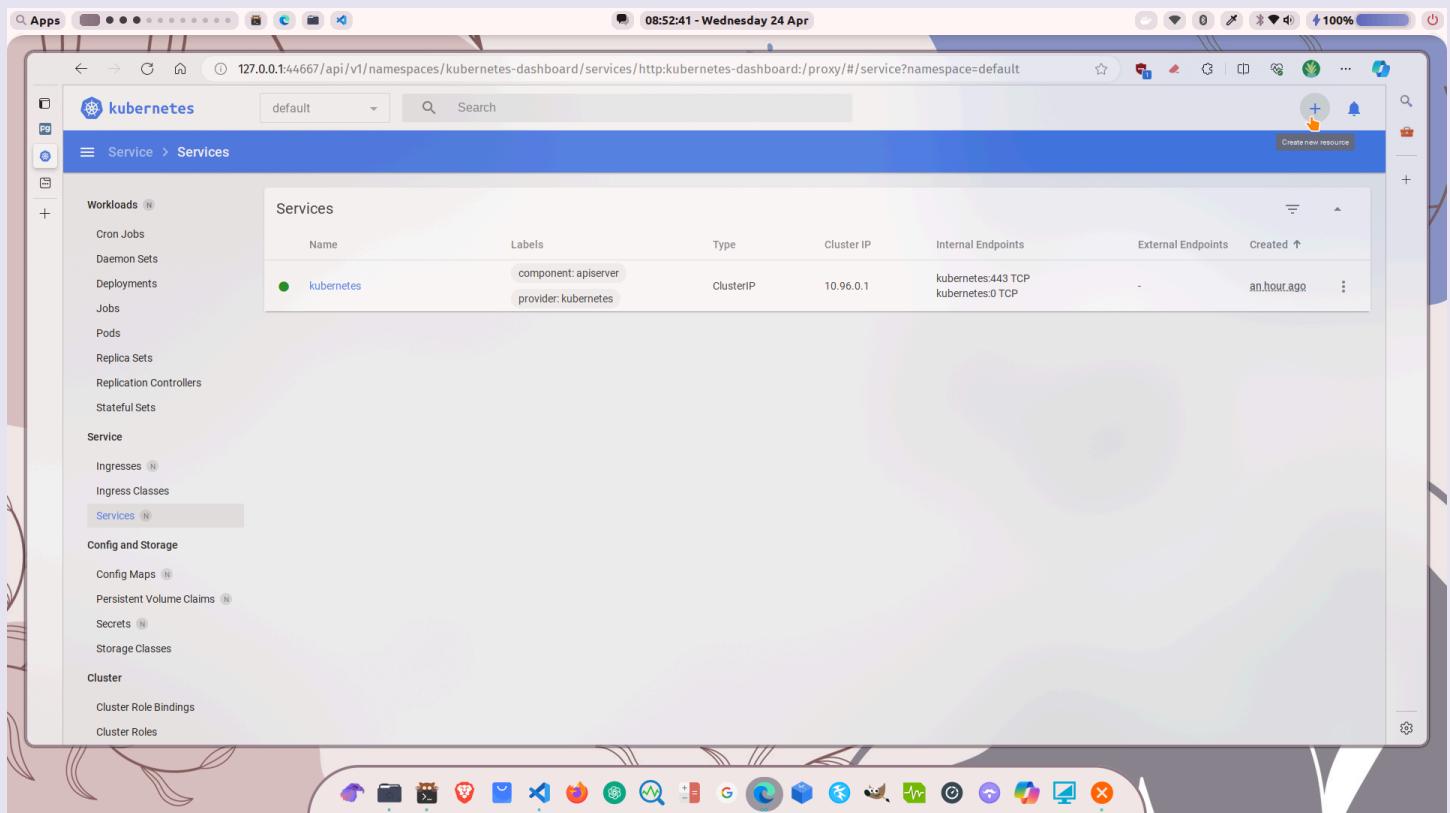
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

e) Now, create service using service file



The screenshot shows the Kubernetes Dashboard interface running in a web browser. The URL in the address bar is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/service?namespace=default`. The dashboard has a sidebar on the left with categories like Workloads, Service, Config and Storage, and Cluster. Under the Service category, the 'Services' tab is selected. The main content area displays a table of services. There is one entry in the table:

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	an hour ago

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

The screenshot shows the Kubernetes Dashboard interface on a Mac OS X desktop. The URL in the address bar is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy#/create?namespace=default`. The title bar indicates the time is 08:52:52 on Wednesday, April 24. The main area is titled "Create" with a sub-section "Workloads". Under "Workloads", there are links for Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, and Service. The "Service" link is currently selected. A central text area contains a YAML configuration for a Service named "postgres":

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: postgres
5   labels:
6     app: postgres
7 spec:
8   type: NodePort
9   ports:
10    port: 5432
11    selector:
12      app: postgres
```

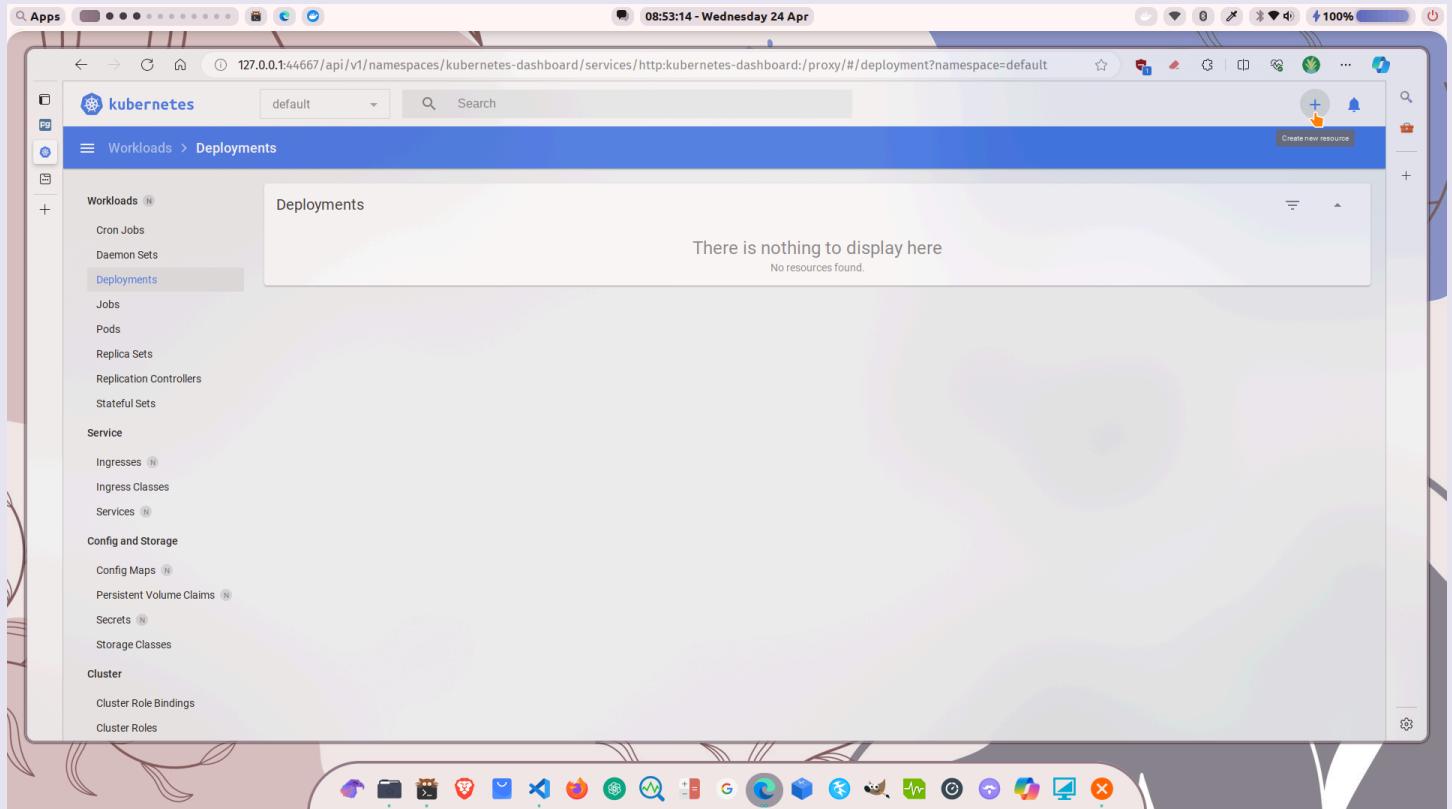
Below the YAML editor are two buttons: "Upload" (highlighted with a red arrow) and "Cancel".

The screenshot shows the Kubernetes Dashboard interface on a Mac OS X desktop. The URL in the address bar is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy#/service?namespace=default`. The title bar indicates the time is 08:53:00 on Wednesday, April 24. The main area is titled "Service > Services". On the left sidebar, under "Service", the "Services" link is selected. The main content area displays a table of services:

Name	Labels	Type	Cluster IP	Internal Endpoints	External Endpoints	Created
postgres	app:postgres	NodePort	10.97.134.180	postgres:5432 TCP postgres:32232 TCP	-	5.seconds.ago
kubernetes	component: apiserver provider: kubernetes	ClusterIP	10.96.0.1	kubernetes:443 TCP kubernetes:0 TCP	-	an.hour.ago

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

f) Finally, create deployment



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 61
EADC Practical 20

The screenshot shows the Kubernetes Dashboard's 'Create' interface for a Deployment. The left sidebar lists various workload types: Cron Jobs, Daemon Sets, Deployments, Jobs, Pods, Replica Sets, Replication Controllers, Stateful Sets, Service, Ingresses, Ingress Classes, Services, Config and Storage, Cluster Role Bindings, and Cluster Roles. The 'Deployments' option is selected. The main area has tabs for 'Create from input', 'Create from file', and 'Create from form'. Below these is a text area for YAML or JSON content, containing the following code:

```
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: postgres
5  spec:
6    replicas: 1
7    selector:
8      matchLabels:
9        app: postgres
10   template:
11     metadata:
12       labels:
13         app: postgres
14     spec:
15       containers:
16         - name: postgres
17           image: postgres:10.1
18           imagePullPolicy: "IfNotPresent"
19 
```

At the bottom of the text area are 'Upload' and 'Cancel' buttons, with 'Upload' being highlighted by a cursor.

The screenshot shows the Kubernetes Dashboard's 'Deployments' list. The left sidebar is identical to the previous screenshot. The main area shows a table titled 'Deployments' with columns: Name, Images, Labels, Pods, and Created. There is one entry:

Name	Images	Labels	Pods	Created
postgres	Postgres:10.1	-	0 / 1	3.seconds.ago

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

g) Check if the pod is created and running successfully

The screenshot shows the Kubernetes Dashboard interface running on a Mac OS X desktop. The URL in the browser is `127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/#/pod?namespace=default`. The dashboard has a sidebar on the left with categories like Workloads, Service, Config and Storage, and Cluster. The main area is titled 'Pods' under 'Workloads'. It lists one pod:

Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created
postgres-7b8bfb4588-hkbd6	postgres:10.1	app:postgres pod-template-hash:7b8bfb4 598	minikube	Running	0	-	-	3.minutes.ago

The pod status is 'Running' and it was created 3 minutes ago. The node is 'minikube'. The pod template hash is '7b8bfb4'.

Name - Yash Lakhtariya

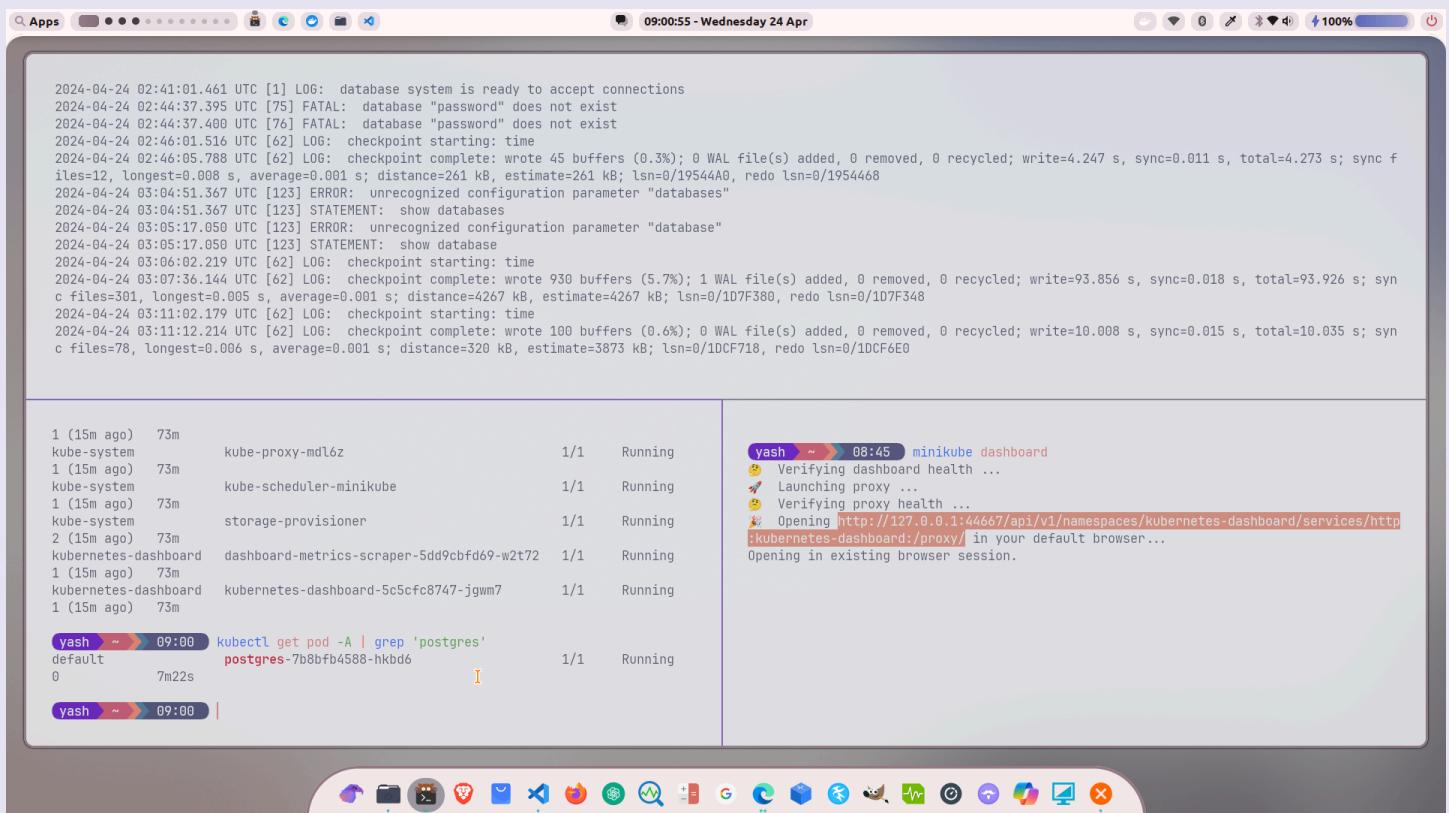
Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

5. Learn to communicate with Kubernetes POD and access Postgres using PgAdmin using Service.

- a) Use kubectl get pod -A command to list all pods and grep to check its output for postgres text containing line, which means the postgres pod is present



The screenshot shows a macOS desktop environment. At the top, there's a menu bar with 'File', 'Edit', 'View', 'Select', 'System', 'Help', and a date/time indicator '09:00:55 - Wednesday 24 Apr'. Below the menu bar is a toolbar with icons for 'Search', 'New', 'Open', 'Save', 'Print', 'Copy', 'Paste', 'Find', 'Replace', and 'Help'. The main window contains two panes. The left pane shows the output of the command 'kubectl get pod -A | grep "postgres"':

```
1 (15m ago) 73m kube-system kube-proxy-mdl6z 1/1 Running
1 (15m ago) 73m kube-system kube-scheduler-minikube 1/1 Running
1 (15m ago) 73m kube-system storage-provisioner 1/1 Running
2 (15m ago) 73m kubernetes-dashboard dashboard-metrics-scraper-5dd9cbfd69-w2t72 1/1 Running
1 (15m ago) 73m kubernetes-dashboard kubernetes-dashboard-5c5cf8747-jgwm7 1/1 Running
1 (15m ago) 73m
```

The right pane shows the output of 'minikube dashboard' with a status message: 'yash ~ 08:45 minikube dashboard'. It includes a link to 'http://127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard/proxy/'. The bottom of the screen features a dock with various application icons.

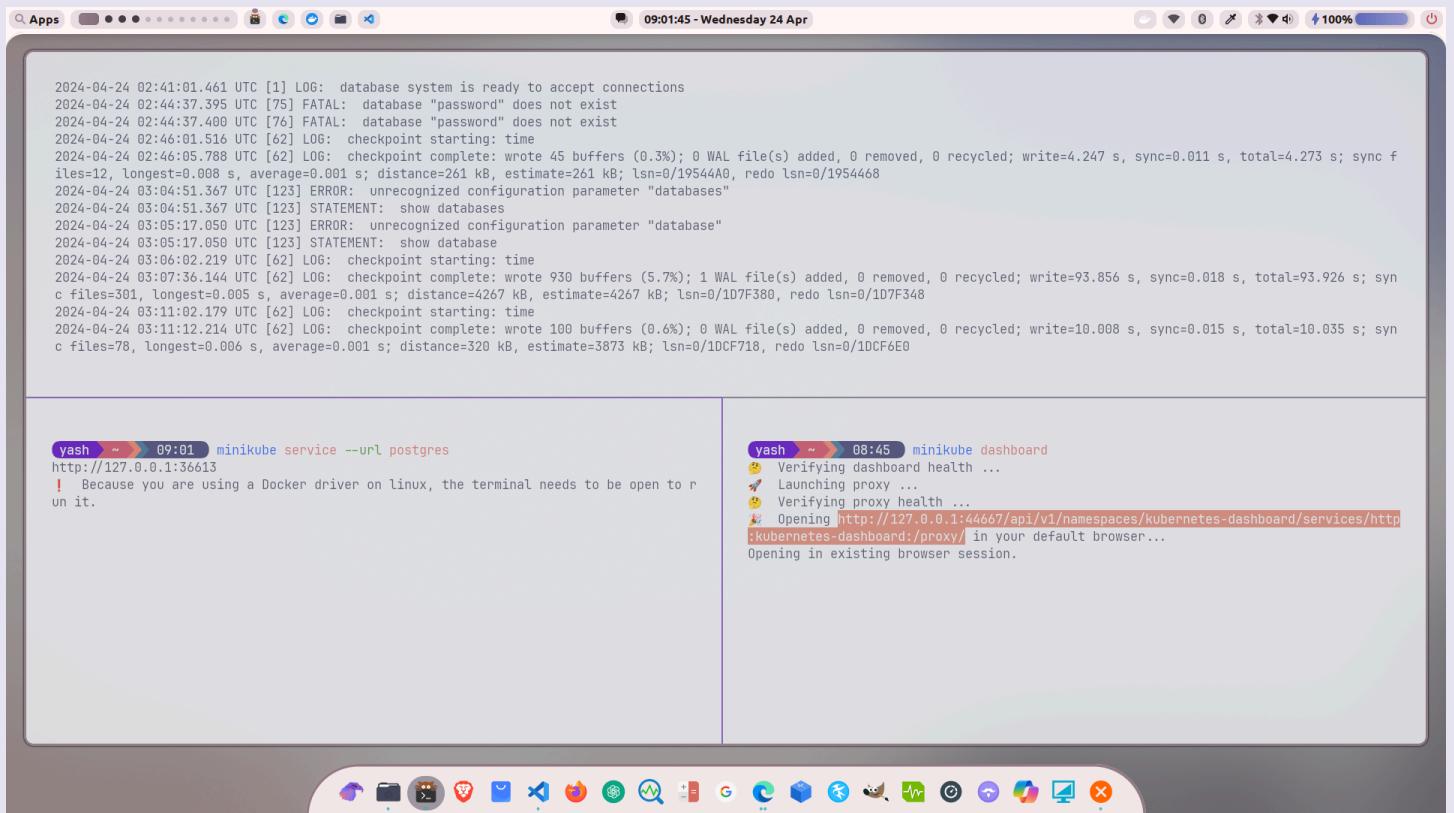
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

- b) Run command, minikube service --url postgres to run postgres service and get its local URL



```
2024-04-24 02:41:01.461 UTC [1] LOG: database system is ready to accept connections
2024-04-24 02:44:37.395 UTC [75] FATAL: database "password" does not exist
2024-04-24 02:44:37.400 UTC [76] FATAL: database "password" does not exist
2024-04-24 02:46:01.531 UTC [62] LOG: checkpoint starting: time
2024-04-24 02:46:05.788 UTC [62] LOG: checkpoint complete: wrote 45 buffers (0.3%); 0 WAL file(s) added, 0 removed, 0 recycled; write=4.247 s, sync=0.011 s, total=4.273 s; sync files=12, longest=0.008 s, average=0.001 s; distance=261 kB, estimate=261 kB; lsn=0/19544A0, redo lsn=0/1954468
2024-04-24 03:04:51.367 UTC [123] ERROR: unrecognized configuration parameter "databases"
2024-04-24 03:04:51.367 UTC [123] STATEMENT: show databases
2024-04-24 03:05:17.050 UTC [123] ERROR: unrecognized configuration parameter "database"
2024-04-24 03:05:17.050 UTC [123] STATEMENT: show database
2024-04-24 03:06:02.219 UTC [62] LOG: checkpoint starting: time
2024-04-24 03:06:02.219 UTC [62] LOG: checkpoint complete: wrote 930 buffers (5.7%); 1 WAL file(s) added, 0 removed, 0 recycled; write=93.856 s, sync=0.018 s, total=93.926 s; sync files=301, longest=0.008 s, average=0.001 s; distance=4267 kB, estimate=4267 kB; lsn=0/1D7F380, redo lsn=0/1D7F346
2024-04-24 03:11:02.179 UTC [62] LOG: checkpoint starting: time
2024-04-24 03:11:12.214 UTC [62] LOG: checkpoint complete: wrote 100 buffers (0.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=10.008 s, sync=0.015 s, total=10.035 s; sync files=78, longest=0.006 s, average=0.001 s; distance=320 kB, estimate=3873 kB; lsn=0/1DCF718, redo lsn=0/1DCF6E0
```



```
yash ~ 09:01 minikube service --url postgres
http://127.0.0.1:36613
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```



```
yash ~ 08:45 minikube dashboard
⌚ Verifying dashboard health ...
🚀 Launching proxy ...
⌚ Verifying proxy health ...
🌐 Opening http://127.0.0.1:44667/api/v1/namespaces/kubernetes-dashboard/services/http://kubernetes-dashboard/proxy/ in your default browser...
Opening in existing browser session.
```

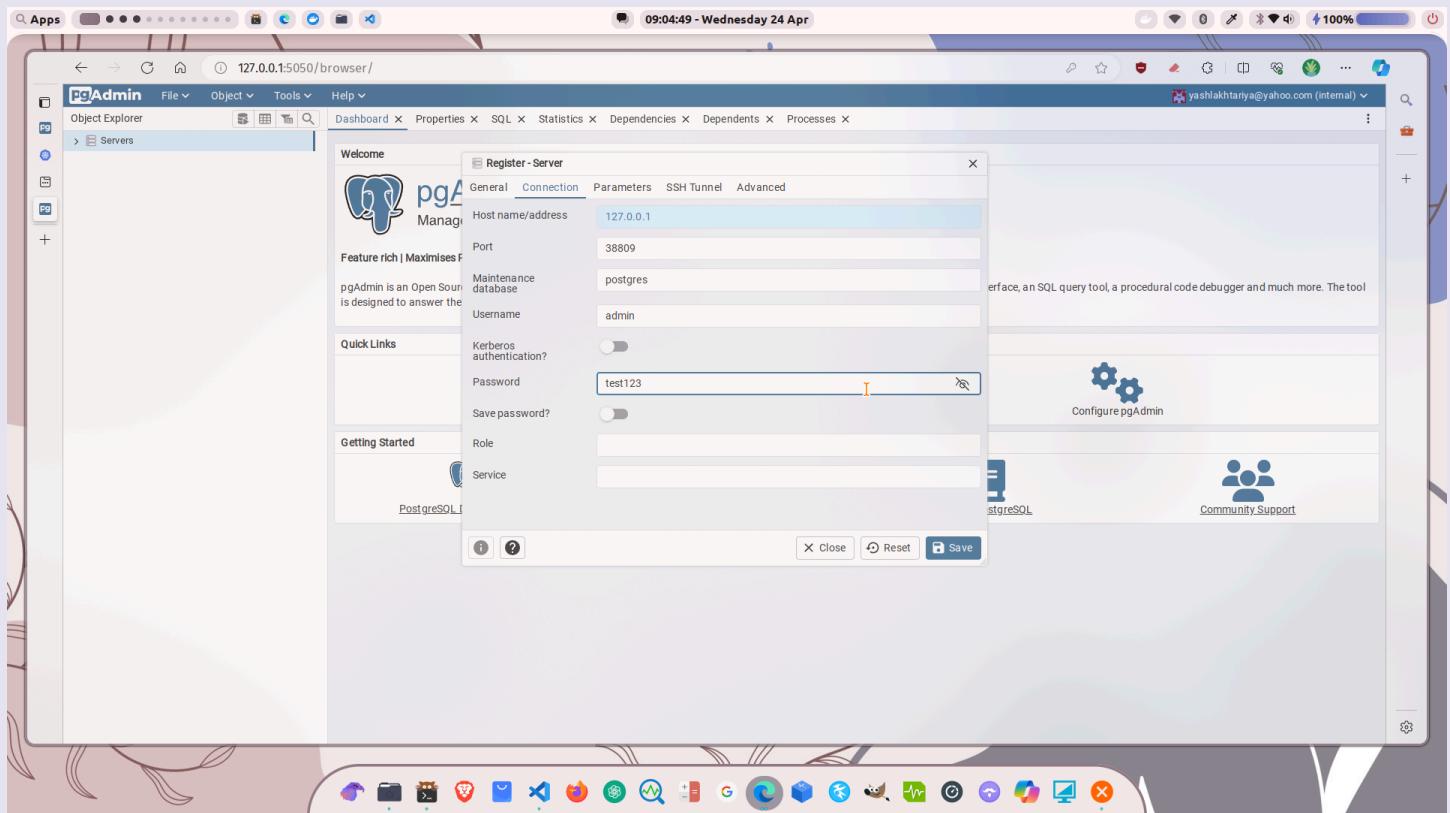
Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

- c) Now, on a local machine's pgadmin, register a server and try connecting to this kubernetes pod of postgres



NOTE : The k8s pod is using container of docker, but is running on localhost or 127.0.0.1 so pgadmin installation of local machine must be used to connect to it, not pgadmin container, which can only connect to other container's postgres

Name - Yash Lakhtariya

Enrollment number - 21162101012

Branch - CBA Batch - 61

EADC Practical 20

- d) The communication between k8s pod of postgres and localhost pgadmin is successful

