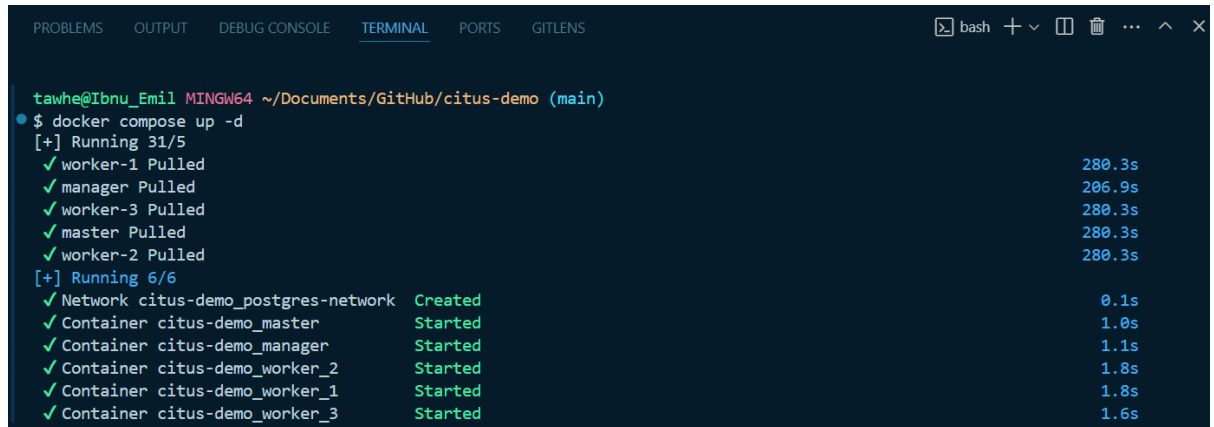


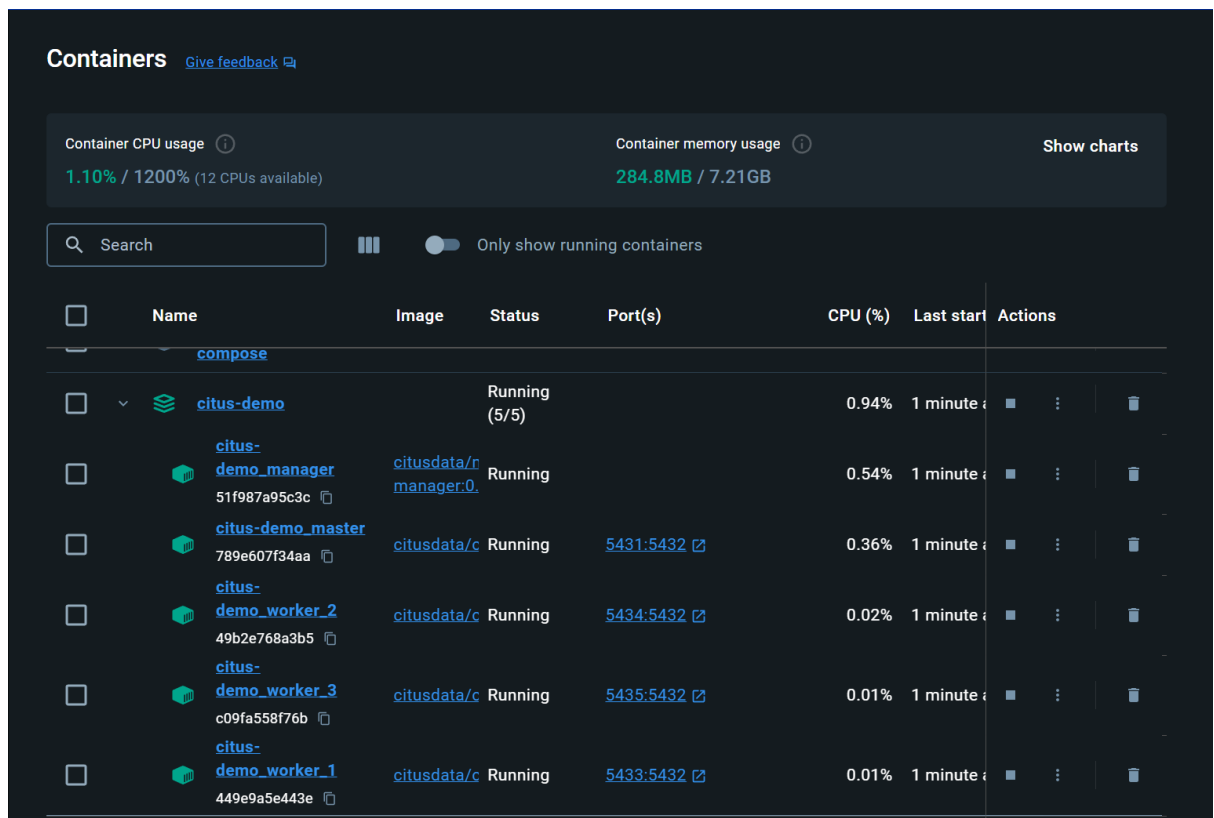
Part 2 – Columnar Database

1. Jalankan Citus di komputer lokal dengan menggunakan docker compose!
 - 1) Langkah pertama jalankan docker compose pada direktori docker-compose.yml dengan mengetik perintah “docker compose up -d”.



```
tawhe@Ibnu_Emil MINGW64 ~/Documents/GitHub/citus-demo (main)
$ docker compose up -d
[+] Running 31/5
  ✓ worker-1 Pulled                                280.3s
  ✓ manager Pulled                                206.9s
  ✓ worker-3 Pulled                                280.3s
  ✓ master Pulled                                 280.3s
  ✓ worker-2 Pulled                                280.3s
[+] Running 6/6
  ✓ Network citus-demo_postgres-network Created      0.1s
  ✓ Container citus-demo_master Started              1.0s
  ✓ Container citus-demo_manager Started             1.1s
  ✓ Container citus-demo_worker_2 Started            1.8s
  ✓ Container citus-demo_worker_1 Started            1.8s
  ✓ Container citus-demo_worker_3 Started            1.6s
```

Berikut adalah tampilan bahwa sudah berhasil menjalankan perintah “docker compose up -d”



Containers [Give feedback](#)

Container CPU usage: 1.10% / 1200% (12 CPUs available) | Container memory usage: 284.8MB / 7.21GB | [Show charts](#)

Search: Only show running containers: ☐

<input type="checkbox"/>	Name	Image	Status	Port(s)	CPU (%)	Last start	Actions
<input type="checkbox"/>	compose						
<input type="checkbox"/>	citus-demo		Running (5/5)		0.94%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	citus-demo_manager 51f987a95c3c	citusdata/manager:0	Running		0.54%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	citus-demo_master 789e607f34aa	citusdata/c	Running	5431:5432	0.36%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	citus-demo_worker_2 49b2e768a3b5	citusdata/c	Running	5434:5432	0.02%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	citus-demo_worker_3 c09fa558f76b	citusdata/c	Running	5435:5432	0.01%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/>	citus-demo_worker_1 449e9a5e443e	citusdata/c	Running	5433:5432	0.01%	1 minute	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Pada aplikasi Docker juga sudah ada container yang sudah dijalankan tadi. Berikut ini adalah bukti bahwa docker compose sudah berhasil.

- 2) Kemudian, langkah berikutnya adalah membuat connection ke PostgreSQL menggunakan DBeaver dengan konfigurasi seperti di bawah ini :

Connect to a database

Connection Settings

PostgreSQL connection settings

Main PostgreSQL Driver properties SSH SSL + Network configurations...

Server

Connect by: ☒ Host ☐ URL

URL: jdbc:postgresql://localhost:5431/store

Host: localhost Port: 5431

Database: store ☐ Show all databases

Authentication

Authentication: Database Native

Username: postgres

Password: ☒ Save password

Advanced

Session role: Local Client: PostgreSQL 16

[Connection variables information](#) [Database documentation](#) Connection details (name, type, ...)

Driver name: PostgreSQL [Driver Settings](#) [Driver license](#)

[Test Connection ...](#) [< Back](#) [Next >](#) [Finish](#) [Cancel](#)

Connection test

Connected (759 ms)

Server: PostgreSQL 15.3 (Debian 15.3-1.pgdg120+1)
PostgreSQL 15.3 (Debian 15.3-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit

Driver: PostgreSQL JDBC Driver 42.7.2

[OK](#) [Details >>](#)

Berikut merupakan hasil connection test dan connection sudah berhasil dibuat.

2. Tuliskan perintah untuk membuat tabel biasa dan tabel columnar!

1) Tabel Columnar

```
create table events_columnar (  
  device_id bigint,  
  event_id bigserial,  
  event_time timestamptz default now (),  
  data jsonb not null  
)  
using columnar;
```

2) Tabel Biasa

```
-- create a row-based table to compare  
create table events_row as select * from events_columnar;
```

3. Masukkan 100.000 baris data ke dalam tabel biasa dan tabel columnar!

```
-- insert some data  
insert into events_columnar (device_id, data)  
select d, '{"hello":"columnar"}' from generate_series(1,100000) d;
```

4. Tampilkan perbedaan ukuran antara tabel biasa dan tabel columnar!

Tables	
events_columnar	288K
events_row	8.1M

Berikut ini adalah perbedaan ukuran yang sangat signifikan dari table_columnar dan table events_row.

5. Tuliskan kesimpulannya!

Kesimpulan dari tabel columnar dan tabel biasa yang sudah dijalankan diatas, yaitu jika menggunakan tabel columnar dapat menyimpan file dengan lebih efisien dibandingkan dengan menggunakan tabel biasa, karena tabel columnar menyimpan secara terpisah, sehingga jika ada 1 juta baris, hanya kolom yang relevan yang disimpan 1 juta kali. Sedangkan jika tabel biasa setiap baris menyimpan semua kolom data, sehingga jika ada 1 juta baris, semua kolom disimpan 1 juta kali.

Jadi tabel columnar dapat menyimpan data dengan jumlah yang sama dengan data yang disimpan pada tabel biasa, tetapi hanya memakan storage yang lebih sedikit dibandingkan tabel biasa, sehingga penyimpanan dapat lebih efisien.