how to install odoo 16 service on ubuntu 24.04 odoo user=$USER=odoouser1 docker-compose.yml with port: 8099 and postgresql user=$USER=odoouser1, and I want to including docker volume in docker-compose.yml file?

To install Odoo 16 as a service on Ubuntu 20.04, create a Dockerized setup using Docker Compose, and include a PostgreSQL instance with a Docker volume, you can follow these steps. This setup assumes you have Docker and Docker Compose installed on your Ubuntu 20.04 server.

### Step 1: Prepare the Environment

Make sure you have Docker and Docker Compose installed on your Ubuntu 20.04 server. If not, you can install them following the official Docker documentation.

### Step 2: Create a Docker Compose File

Create a `docker-compose.yml` file in your project directory. This file will define your Odoo and PostgreSQL services along with the necessary configurations:

```yaml

version: '3.8'

services:

postgres:

image: postgres:12

container\_name: postgres-container

environment:

POSTGRES\_USER: odoouser1

POSTGRES\_PASSWORD: your\_postgres\_password

POSTGRES\_DB: odoo\_db

volumes:

- postgres-data:/var/lib/postgresql/data

odoo:

image: odoo:16

container\_name: odoo-container

ports:

- "8099:8069"

environment:

DB\_HOST: postgres

DB\_PORT: 5432

DB\_USER: odoouser1

DB\_PASSWORD: your\_postgres\_password

DB\_NAME: odoo\_db

volumes:

- odoo-data:/var/lib/odoo

volumes:

postgres-data:

odoo-data:

```

### Explanation:

- \*\*postgres Service\*\*:

- Uses the `postgres:12` Docker image.

- Sets environment variables (`POSTGRES\_USER`, `POSTGRES\_PASSWORD`, `POSTGRES\_DB`) for PostgreSQL.

- Defines a volume `postgres-data` to persist PostgreSQL data (`/var/lib/postgresql/data`).

- \*\*odoo Service\*\*:

- Uses the `odoo:16` Docker image.

- Maps port `8099` from the container to port `8069` on the host.

- Sets environment variables (`DB\_HOST`, `DB\_PORT`, `DB\_USER`, `DB\_PASSWORD`, `DB\_NAME`) for Odoo to connect to PostgreSQL.

- Defines a volume `odoo-data` to persist Odoo data (`/var/lib/odoo`).

### Step 3: Create User Directories

Create directories on your host system (`/home/odoouser1/postgres-data` and `/home/odoouser1/odoo-data`) where the Docker volumes will be mounted. Replace `/home/odoouser1` with your actual user's home directory.

```bash

mkdir -p /home/uofdock/postgres-data

mkdir -p /home/uofdock/odoo-data

```

### Step 4: Run Docker Compose

Navigate to the directory containing your `docker-compose.yml` file and run:

```bash

docker-compose up -d

```

This command will start the PostgreSQL and Odoo containers in detached mode (`-d`), based on the configurations specified in `docker-compose.yml`. The `-d` flag runs containers in the background.

### Step 5: Access Odoo

Once both containers are running, you should be able to access Odoo at `http://localhost:8099`.

### Notes:

- \*\*Permissions\*\*: Ensure that the directories (`/home/odoouser1/postgres-data` and `/home/odoouser1/odoo-data`) are writable by the user running Docker (typically requires sudo or root access).

- \*\*Security\*\*: Use strong passwords and ensure proper firewall rules if exposing ports (`8099` for Odoo) to the internet.

This setup provides a basic Dockerized environment for Odoo 16 with PostgreSQL, using Docker Compose to manage both services and volumes for persistent data storage. Adjust configurations (`image versions`, `environment variables`, etc.) based on your specific requirements and deployment environment.