Nextcloud Setup and Project Report

Ryan Fitzgerald, May 5th 2025

Prerequisites

In order to set up Nextcloud successfully, the following systems/apps need to be installed.

- **Docker/Docker Engine**
- **Docker compose**
- **Text editor (VIM, Nano, etc.)**
- **An internet browser (Firefox, Chrome, etc.)**

For this project, Docker is used to run the Nextcloud and MariaDB containers. It creates the environment that the application and its database will be running in. Docker Compose is used to define and manage these two docker instances using a docker-compose.yml file. A text editor is needed in order to configure the docker-compose.yml file. In order to access the Nextcloud interface, we will need to access it using an internet browser.

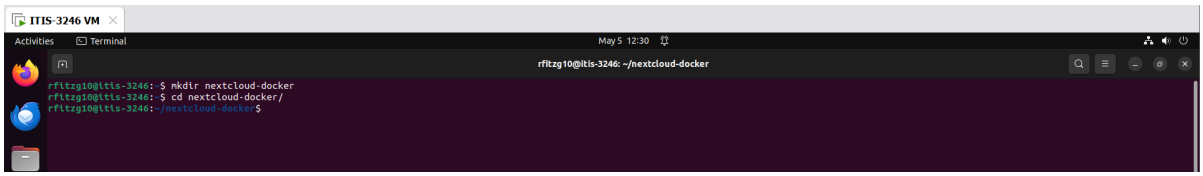
Project Overview

The goal of this project as outlined in the project proposal is to set up and configure a Nextcloud instance using Docker locally. I'll be setting up Nextcloud to be connected to a MariaDB database and setting up the database via a configured docker-compose file. Once the setup in Docker is complete, the final setup of Nextcloud will take place through the web interface, which we will be accessing through Firefox in this case. I'll be doing this project on a Virtual Machine, but that is not necessary for any of the steps.

Step 1: Creating New Directory

While not completely necessary, in order to keep everything in one place that is easy to access, we will be creating a new directory that will store the docker-compose file. To do this, we will simply run the mkdir command, and give the new directory an appropriate name. We will then change to the new directory.

```
mkdir <new directory>
cd <new directory>
```



Step 2: Configuring docker-compose.yml File

In order to get Nextcloud up and running, we'll need to create and configure a docker-compose.yml file that will set up the two containers we will be utilizing, which is Nextcloud with a MariaDB database backend. While in the new directory, run the command to start editing a new file with whichever text editor you are using.

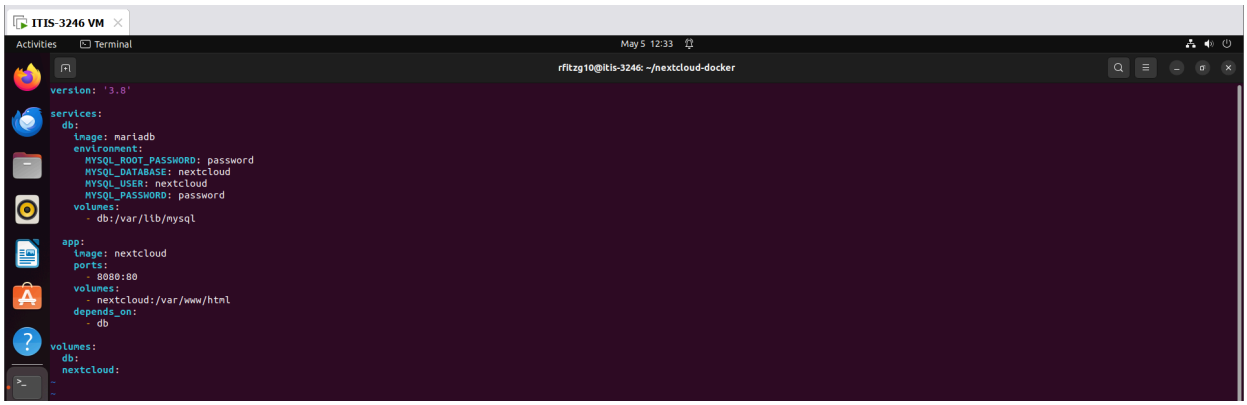
```
vim docker-compose.yml
nano docker-compose.yml
```

Now that we are in the text editor, we can start configuring it for what we need. In this case I already have made the configured file, and you can copy and paste it into the text editor.

```
version: '3.8'

services:
  db:
     image: mariadb
     environment:
     MYSQL_ROOT_PASSWORD: password
     MYSQL_DATABASE: nextcloud
     MYSQL_USER: nextcloud
     MYSQL_PASSWORD: password
     volumes:
     - db:/var/lib/mysql

  app:
     image: nextcloud
     ports:
     - 8080:80
     volumes:
     - nextcloud:/var/www/html
     depends_on:
     - db

volumes:
  db:
  nextcloud:
```

To break down the contents of the file, the first line specifies which version of the Compose file format to use. The services section defines the contains that Docker Compose will be managing for our instance of Nextcloud to function. It creates a container from the official MariaDB image, which is the database we are using. The environment section has the variables needed to configure the database. It sets the root password, creates a new database named nextcloud, creates a new user, and gives that user a password. You can rename these as needed, but for

this project I'll be keeping it simple. The volumes line creates a new volume so the database is saved even if the container is stopped or deleted. The app section creates a new container from the official Nextcloud image. It then maps the 8080 port on the host machine to 80 inside the container, which allows us to access it via a browser. As with the services section, the volumes line creates a new volume so the data for the Nextcloud image is saved should anything happen to the container. The depends_on section ensures that the database is up and running before the Nextcloud app starts. The last volumes section gives the two made volumes the names db and nextcloud.

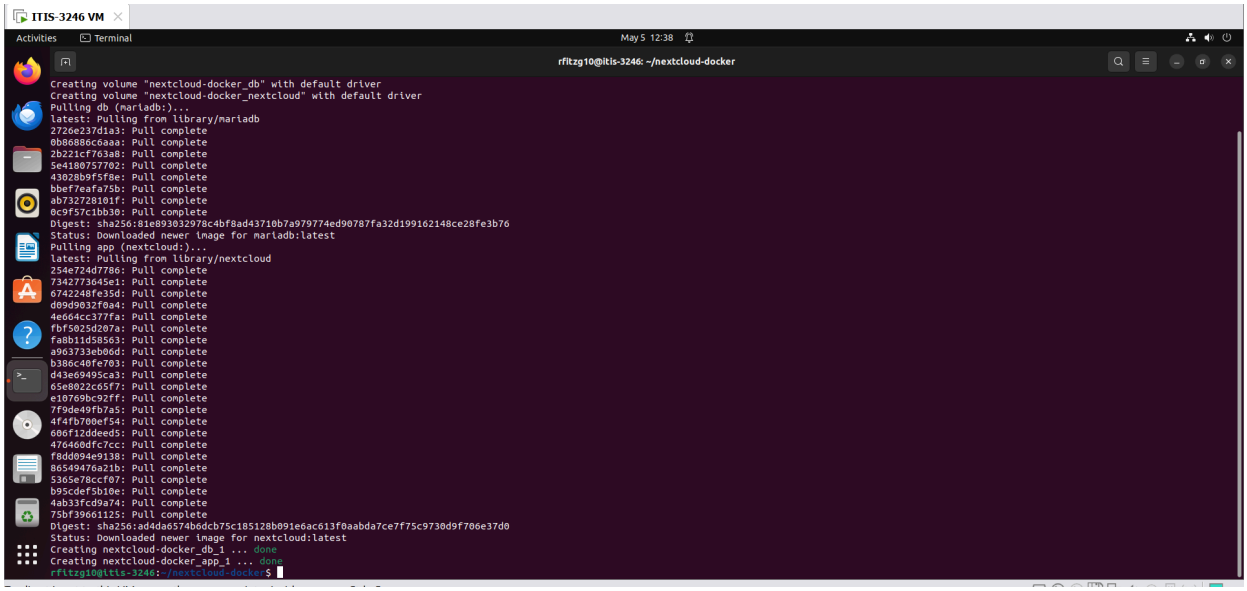


Step 3: Starting the Containers

Now that we have the compose file configured, we can now start the containers by running this command.

```
sudo docker-compose up -d
```

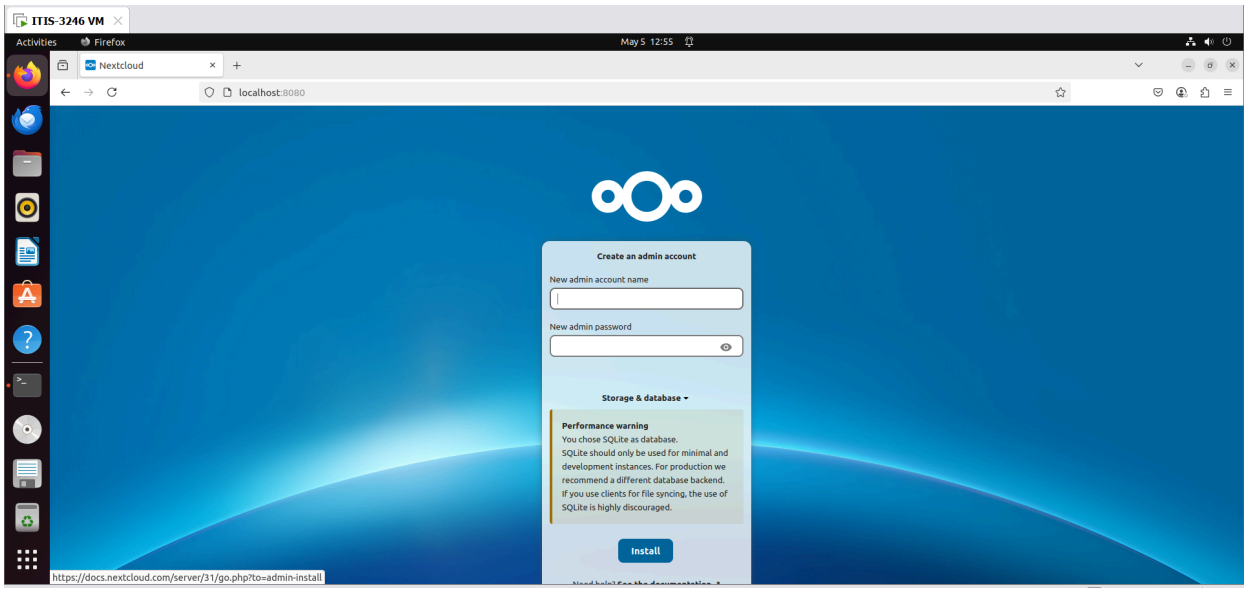
This command will download the needed images for MariaDB and Nextcloud, create the two containers for them, and start them based on our configurations in the compose file.



Step 4: Getting to Nextcloud Web Interface

Once the containers have been successfully created, we can now open up our browser of choice. In the address bar, enter the localhost address if you are using the same machine we set the containers up on, or use the IP address of the machine running Docker.

```
localhost:8080
http://<ip_address>:8080/
```



Step 5: Final Setup and Installation

You should now see the initial Nextcloud setup page in the browser. Enter in a new admin account name and a password for it, then click on the ‘Storage & database’ option. The data folder by default should be /var/www/html/data, which is what we will leave it as. Then, click on ‘MySQL/MariaDB’ and fill in the options as follows:

```
Database account: nextcloud
Database password: password
Database name: nextcloud
Database host: db
```

Once you’ve entered everything correctly, click on ‘Install’ and the installation will start. It may take a minute for the first setup. Once it is done, you will get a list of recommended apps. You can choose these if you’d like, but for this project I’ll be skipping them. After that, if you see the ‘Recommended files’ section on your screen, your Nextcloud instance is all set up and ready to go! You can navigate to the ‘Files’ page at the top left. From there, you upload and download files, organize folders, and use Nextcloud’s features.