

Optimalisasi Deployment Aplikasi Cloud Computing Berbasis Website Dengan Pendekatan DevOps Menggunakan OpenMediaVault, Portainer, dan Docker

Saidi Ramadan Siregar¹, Hery Sunandar², Pristiwanto³, Azlan⁴

¹Teknik Informatika, ²Teknologi Informasi, Universitas Budi Darma, ³Informatika, STMIK Mulia Darma,

⁴Sistem Informasi, STMIK Triguna Darma

Email: ¹saidiramadan89@gmail.com, ²herysun1975@gmail.com, ³4nt0.82@gmail.com, ⁴azlansaja19@gmail.com

Email Penulis Korespondensi: saidiramadan89@gmail.com

Article History:

Received Jun 15th, 2025

Revised Jun 30th, 2025

Accepted Jul 30th, 2025

Abstrak

Salah satu kendala yang sering terjadi dalam proses deployment aplikasi cloud computing berbasis website adalah ketidakcocokan (incompatibility) antara requirement aplikasi dengan lingkungan tempat aplikasi dijalankan. Masalah ini bisa mencakup versi web server, database, dependensi pendukung lainnya, serta konfigurasi sistem yang tidak tepat. Akibatnya, aplikasi gagal dideploy secara menyeluruh sehingga tidak dapat digunakan sesuai fungsinya, misalnya sebagai data center layaknya layanan Google Drive. Hal ini tentu menjadi tantangan tersendiri, khususnya bagi tim IT Support di setiap instansi yang bertanggung jawab atas proses deployment aplikasi cloud computing atau berbagai aplikasi website lainnya. Untuk mengatasi kendala dalam proses deployment aplikasi cloud computing, solusi yang dapat diterapkan adalah dengan mengikuti perkembangan teknologi DevOps, misalnya melalui pemanfaatan openmediavault, portainer, dan docker. Ketiga aplikasi ini memungkinkan proses deployment dilakukan dengan cepat, efisien, dan tidak memerlukan waktu yang lama. openmediavault berperan sebagai sistem operasi utama yang digunakan dalam lingkungan server. Portainer berfungsi untuk mempermudah manajemen dan pemetaan port layanan berbasis container melalui antarmuka grafis yang intuitif. Docker digunakan sebagai platform utama untuk menjalankan stack aplikasi dalam bentuk container, sehingga dapat menjamin konsistensi antara lingkungan pengembangan dan produksi. Melalui pendekatan ini, proses deployment tidak hanya menjadi lebih stabil dan terstandarisasi, tetapi juga meminimalisir risiko kesalahan konfigurasi dan ketergantungan sistem yang tidak sesuai. Penerapan solusi berbasis DevOps ini menjadi langkah strategis bagi instansi atau organisasi yang ingin mengelola infrastruktur cloud computing secara modern, fleksibel, dan mudah diintegrasikan dengan layanan digital lainnya.

Kata Kunci : cloud computing, deployment, aplikasi, web, linux, openmediavault, portainer, docker

Abstract

One of the obstacles that often occurs in the deployment process of website-based cloud computing applications is the incompatibility between the application requirements and the environment where the application is run. This problem can include the version of the web server, database, other supporting dependencies, as well as improper system configuration. As a result, the application fails to be fully deployed so that it cannot be used according to its function, for example as a data center like Google Drive service. This is certainly a challenge, especially for the IT Support team in every agency that is responsible for the deployment process of cloud computing applications or various other website applications. To overcome obstacles in the cloud computing application deployment process, the solution that can be applied is to follow the development of DevOps technology, for example through the use of openmediavault, portainer, and docker. These three applications allow the deployment process to be done quickly, efficiently, and does not require a long time. openmediavault acts as the main operating system used in the server environment. Portainer serves to simplify the management and mapping of container-based service ports through an intuitive graphical interface. Docker is used as the main platform to run the application stack in the form of containers, so it can guarantee consistency between the development and production environments. Through this approach, the deployment process not only becomes more stable and standardized, but also minimizes the risk of configuration errors and incompatible system dependencies. The implementation of DevOps-based solutions is a strategic step for agencies or organizations that want to manage cloud computing infrastructure in a modern, flexible, and easily integrated way with other digital services.

Keyword : cloud computing, deployment, aplikasi, web, linux, openmediavault, portainer, docker

1. PENDAHULUAN

Perkembangan teknologi informasi yang pesat telah mendorong transformasi digital di berbagai sektor, termasuk dalam pengelolaan data dan layanan berbasis website. Salah satu pendekatan yang banyak digunakan saat ini adalah cloud computing, yang memungkinkan akses data dan aplikasi secara fleksibel melalui jaringan internet. Cloud computing menawarkan berbagai keunggulan, seperti efisiensi sumber daya, skalabilitas, dan kemudahan integrasi dengan berbagai layanan digital.

Namun demikian, proses deployment atau penyebaran aplikasi cloud computing berbasis website seringkali menghadapi berbagai tantangan teknis. Salah satu permasalahan utama yang kerap terjadi adalah ketidakcocokan (incompatibility) antara kebutuhan teknis aplikasi (requirement) dengan lingkungan sistem yang digunakan. Ketidakesesuaian versi web server, database, library pendukung, serta konfigurasi sistem yang kurang tepat dapat menyebabkan kegagalan dalam proses deployment. Akibatnya, aplikasi tidak dapat berfungsi secara optimal atau bahkan tidak dapat dijalankan sama sekali. Hal ini menjadi perhatian serius bagi tim IT Support, khususnya di lingkungan institusi atau organisasi yang bergantung pada layanan berbasis web sebagai pusat data atau sistem informasi.

Untuk mengatasi permasalahan tersebut, diperlukan pendekatan yang lebih modern dan terintegrasi. Salah satu solusi yang kini banyak diterapkan adalah konsep DevOps, yang menggabungkan praktik pengembangan perangkat lunak (development) dan operasional sistem (operations) secara kolaboratif. Dengan memanfaatkan teknologi seperti OpenMediaVault, Portainer, dan Docker, proses deployment dapat dilakukan secara lebih cepat, efisien, dan terstandarisasi. Teknologi ini tidak hanya menyederhanakan manajemen server dan aplikasi, tetapi juga menjamin konsistensi lingkungan dari tahap pengembangan hingga produksi.

Pendahuluan ini menjadi landasan penting dalam mengkaji bagaimana penerapan teknologi DevOps melalui integrasi OpenMediaVault, Portainer, dan Docker mampu menjawab tantangan dalam deployment aplikasi cloud computing berbasis website, serta memberikan solusi praktis yang dapat diimplementasikan di berbagai lingkungan organisasi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Metode yang digunakan dalam penelitian ini bersifat studi kasus implementatif, dengan langkah-langkah sebagai berikut:

- a. Merancang Infrastruktur
- b. Menginstalasi openmediavault
- c. Menginstalasi omv-extras
- d. Menginstalasi portainer
- e. Deploy dua stack docker aplikasi nextcloud dan jellyfin
- f. Mengevaluasi deployment

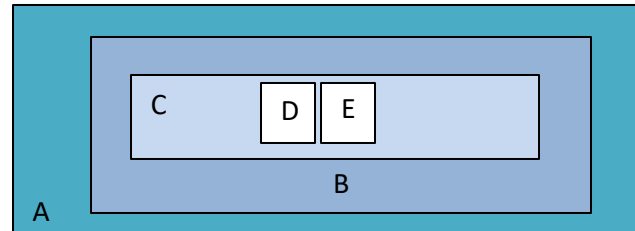
2.2 Requirement

Cloud computing adalah model komputasi berbasis internet yang memungkinkan pengguna mengakses sumber daya komputasi (seperti server, penyimpanan, dan aplikasi) sesuai kebutuhan, tanpa harus memiliki dan mengelola infrastruktur secara langsung. Layanan cloud seperti *google drive* atau *dropbox* adalah contoh pemanfaatan *cloud computing* dalam bentuk Software as a Service (SaaS)[1][2]. DevOps merupakan pendekatan kolaboratif antara tim pengembang (developer) dan tim operasional (operations) yang bertujuan untuk mempercepat penyebaran perangkat lunak, meningkatkan efisiensi, serta menjaga kualitas layanan. DevOps mengedepankan otomatisasi, integrasi berkelanjutan, dan manajemen infrastruktur berbasis kode [3][4]. Docker adalah platform open-source yang memungkinkan pengemasan aplikasi beserta semua dependensinya ke dalam sebuah container yang ringan dan portabel. Docker menjamin konsistensi lingkungan antara pengembangan, pengujian, dan produksi [5][6][7]. Portainer adalah antarmuka grafis berbasis web untuk mengelola container Docker secara mudah dan efisien. Dengan Portainer, pengguna dapat membuat, menghapus, serta memonitor kontainer tanpa perlu menggunakan perintah baris (CLI) [8]. OMV dapat dikombinasikan dengan Docker untuk membangun server aplikasi berbasis cloud yang efisien dan hemat biaya [9]. OpenMediaVault (OMV) adalah sistem operasi berbasis Debian Linux yang dirancang khusus sebagai platform Network Attached Storage (NAS)[10]. Nextcloud Files adalah perangkat lunak penyimpanan cloud dan berbagi file yang menyediakan akses mudah untuk berbagi dan kolaborasi dari mana saja, kapan saja. Semua itu tanpa kebocoran data ke pihak ketiga dan dengan kontrol penuh atas data [11]. Jellyfin adalah perangkat lunak media open-source yang memungkinkan pengguna mengelola dan melakukan streaming koleksi media pribadi secara mandiri. Platform ini menjadi alternatif bebas dari layanan seperti Emby dan Plex yang bersifat tertutup. Jellyfin dikembangkan dari versi Emby 3.5.2 dan telah dipindahkan ke kerangka kerja .NET Core agar dapat berjalan di berbagai sistem operasi. [12].

3. HASIL DAN PEMBAHASAN

3.1 Merancang Infrastruktur

Model infrasktruktur yang akan direncanakan pada penelitian ini ditampilkan pada gambar berikut.

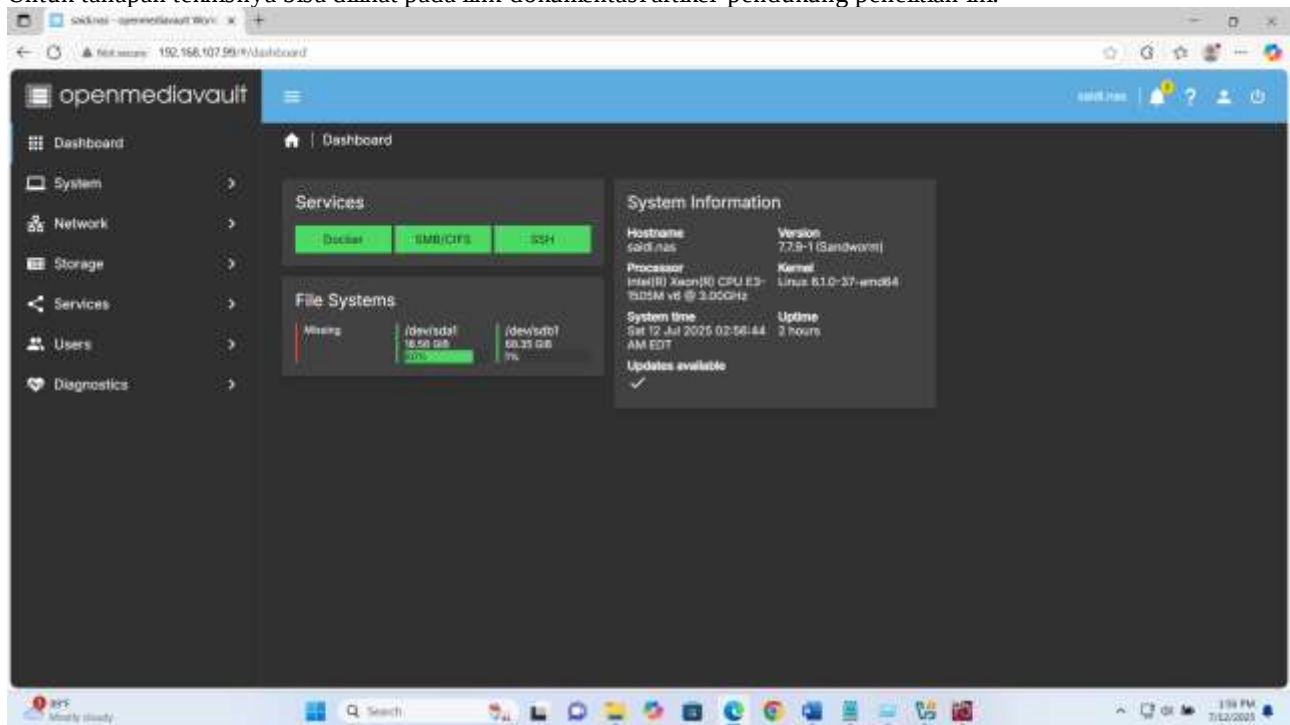


Gambar 1. Infrastruktur

Pada gambar di atas A adalah sistem operasi utama yakni openmediavault, B adalah portainer, C adalah docker sedangkan D dan E aplikasi yang berjalan dalam lingkungan docker. Jadi system operasi utama yang digunakan adalah openmediavault berbasis linux-debian yang akan mengatur yang ada didalamnya termasuk portainer dan docker. Openmediavault juga disebut *Network Address Storage (NOS)* untuk manajemen storage dan layanan jaringan

3.1.2 Menginstalasi openmediavault

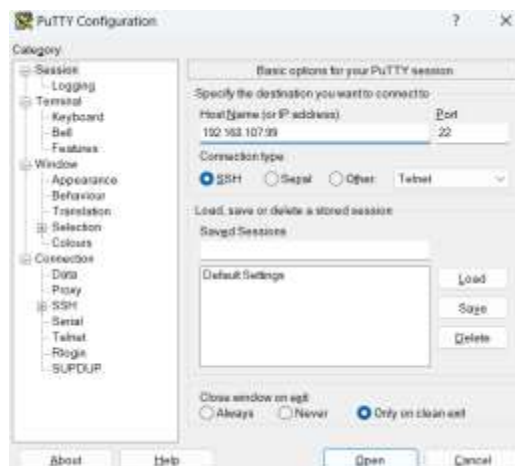
Openmediavault didapatkan dari situs resminya <https://www.openmediavault.org/download.html> kemudian dijalankan pada software virtualbox dengan mengakses alamat ip server openmediavault. Berikut tampilan server openmediavault berbasis linux-debian. Dalam melakukan pemasangan ada beberapa aplikasi pendukung yang dibutuhkan misalnya virtualbox, iso image, kemudian pengaturan-pengaturan yang harus dilakukan pada aplikasi tersebut sehingga proses pemasangan openmediavault sesuai dengan harapan. Berikut tampilan openmediavault setelah berhasil dipasang. Untuk tahapan teknisnya bisa dilihat pada link dokumentasi artikel pendukung penelitian ini.



Gambar 2. Tampilan openmediavault

3.1.3 Menginstalasi omv-extras

- Akses OMV via SSH (misalnya dengan PuTTY) atau langsung di terminal (jika pakai VirtualBox atau langsung di mesin fisik).

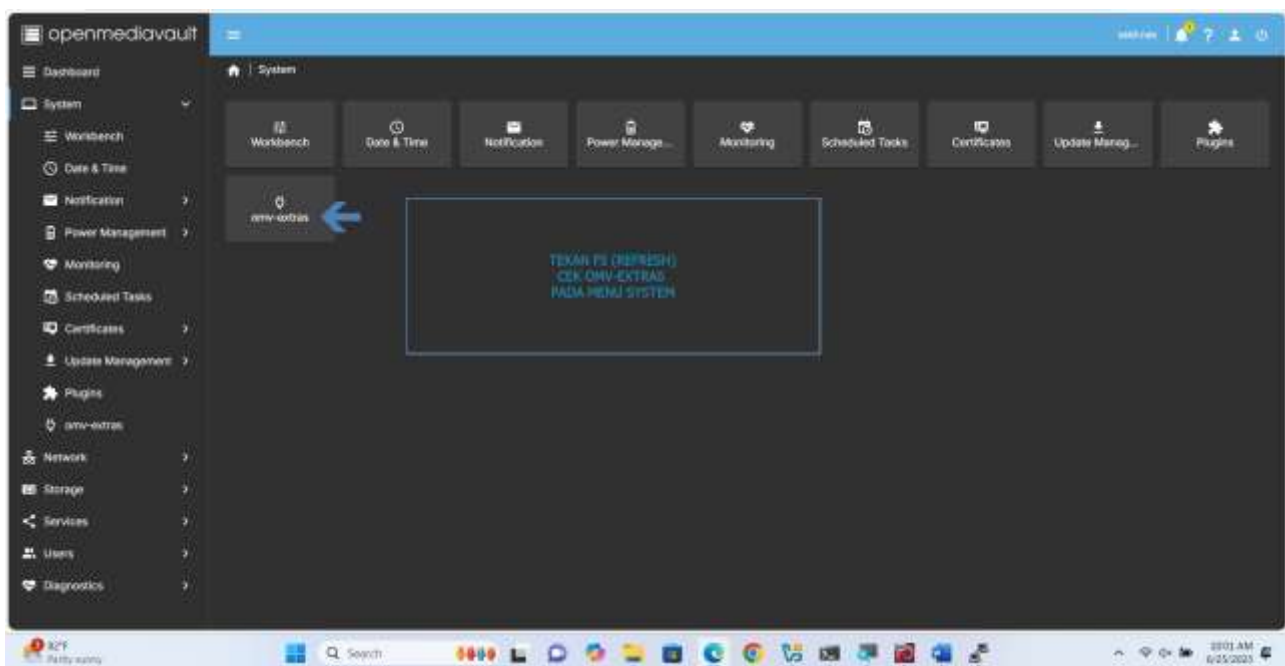


Gambar 3. Remote Server

b. Jalankan perintah berikut untuk menginstall OMV-Extras (pastikan koneksi internet aktif):

```
root@saidi:~# wget -O - https://github.com/OpenMediaVault-Plugin-Developers/packages/raw/master/install | bash
```

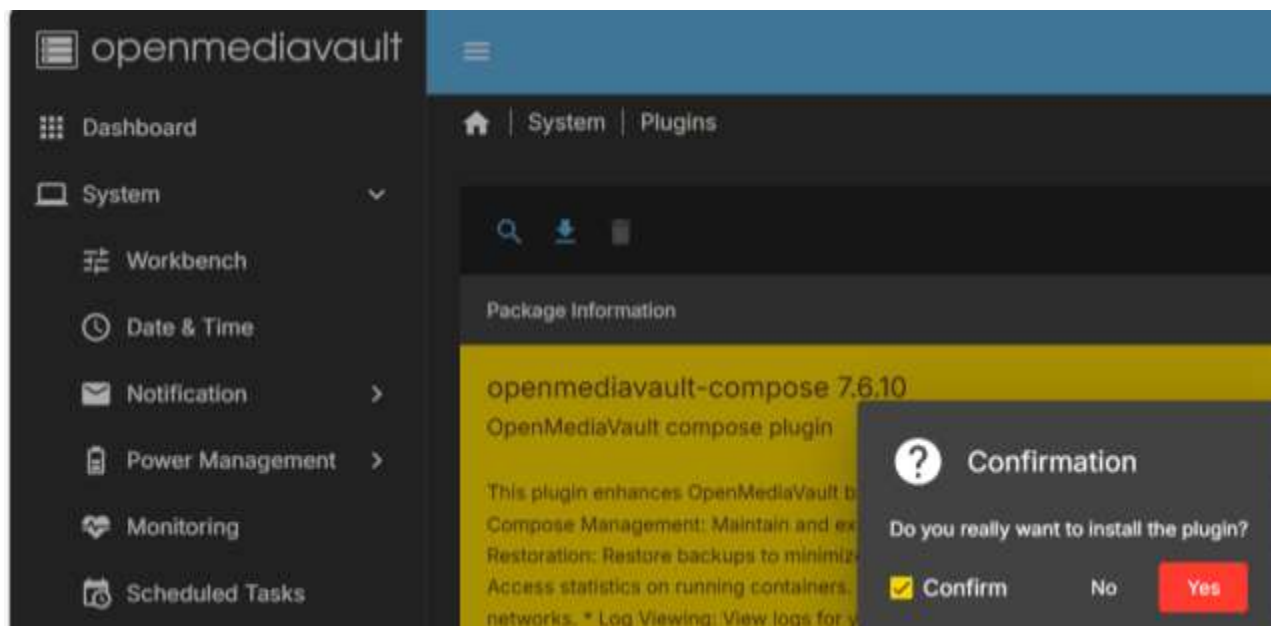
Gambar 4. Perintah untuk melakukan download dan instalasi paket omv-extras



Gambar 5. Omv-extras berhasil terpasang pada ovenmediavault



Gambar 6. Mengaktifkan Docker repo



Gambar 7. Memasang Paket openmediavault-compose 7.6.10

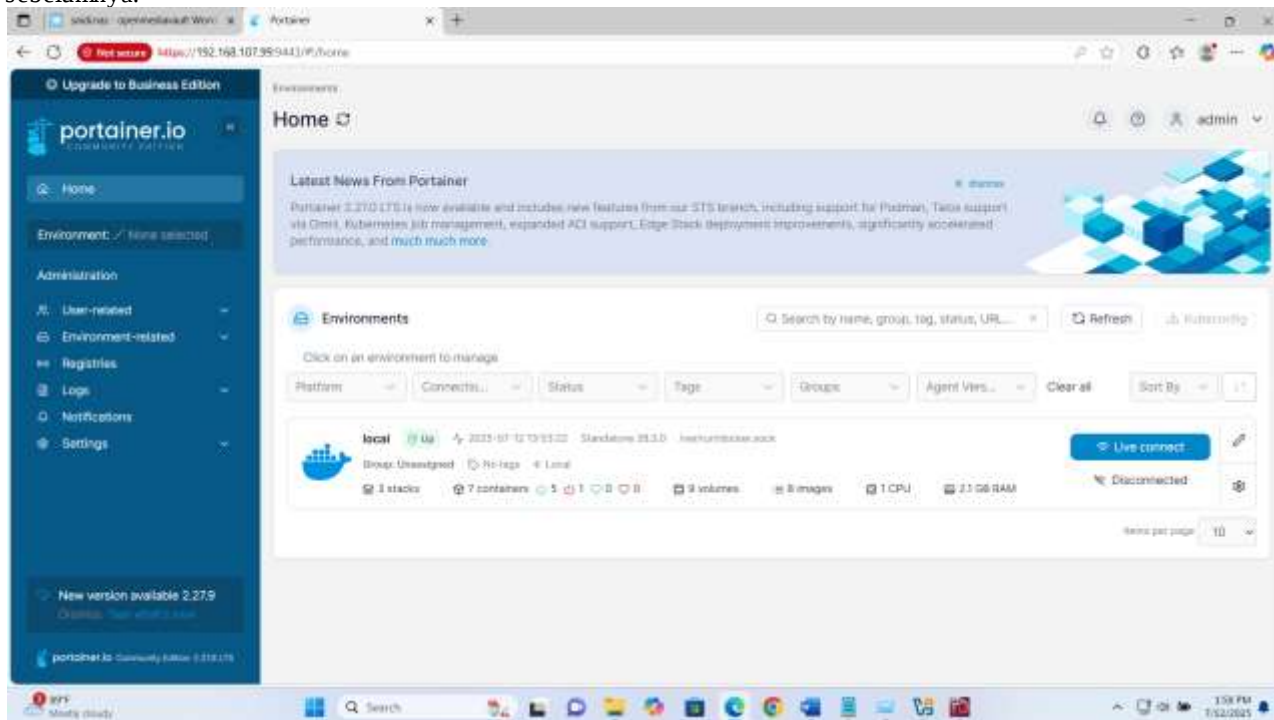
3.1.4 Menginstalasi Portainer

Pemasangan portainer di dalam openmediavault melewati beberapa tahapan seperti melakukan remote ke dalam server, melakukan update apt dan seterusnya. Adapun perintah instalasi yang diset pada terminal putty adalah sebagai berikut

```
root@saiddi:~# docker run -d --p 8080:8080 --p 9443:9443 --name portainer --restart=always --v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
```

Gambar 3.8 Set Perintah Instalasi Portainer

Gambar berikut adalah portainer berhasil dipasang di dalam openmediavault. Dokumentasinya dapat dilihat pada link sebelumnya.



Gambar 8. Tampilan Portainer dan docker otomatis tampil

3.1.5 Deploy Aplikasi Cloud Computing

Aplikasi yang akan didploy adalah nextcloud yang mirip dengan google drive sedangkan jellyfin adalah aplikasi yang mirip dengan youtube. Berikut adalah kode perintah stacknextcloud dan jellyfin yang akan diset dalam docker

version: '3'

services:

db:

image: mariadb

container_name: nextcloud-db

restart: always

environment:

MYSQL_ROOT_PASSWORD: example_root_password

MYSQL_PASSWORD: nextcloudpass

MYSQL_DATABASE: nextcloud

MYSQL_USER: nextclouduser

volumes:

- nextcloud_db:/var/lib/mysql

app:

image: nextcloud

container_name: nextcloud-app

restart: always

ports:

- 8080:80

links:

- db

volumes:

- nextcloud_data:/var/www/html

environment:

MYSQL_PASSWORD: nextcloudpass

MYSQL_DATABASE: nextcloud

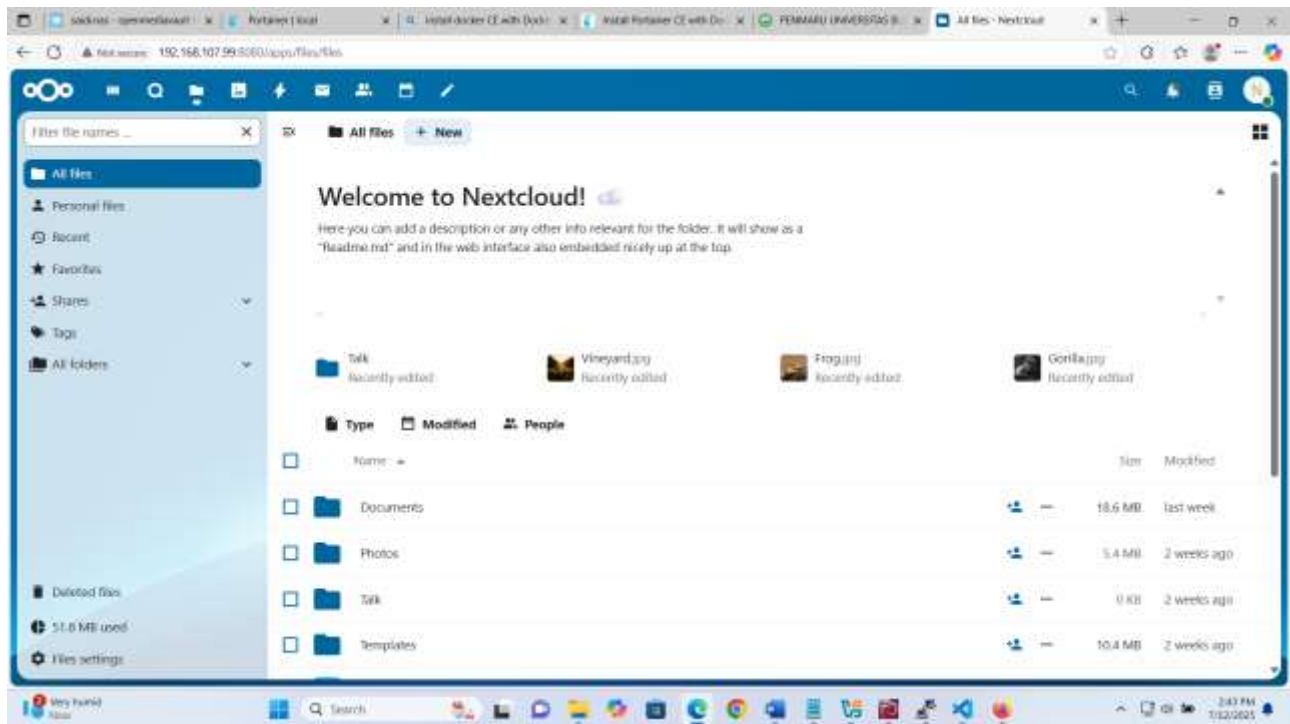
MYSQL_USER: nextclouduser

MYSQL_HOST: db

volumes:

nextcloud_data:

nextcloud_db:



Gambar 9. Tampilan Nextcloud

Berikut kode stack jellyfin

version: "3.3"

services:

jellyfin:

image: jellyfin/jellyfin

container_name: jellyfin

ports:

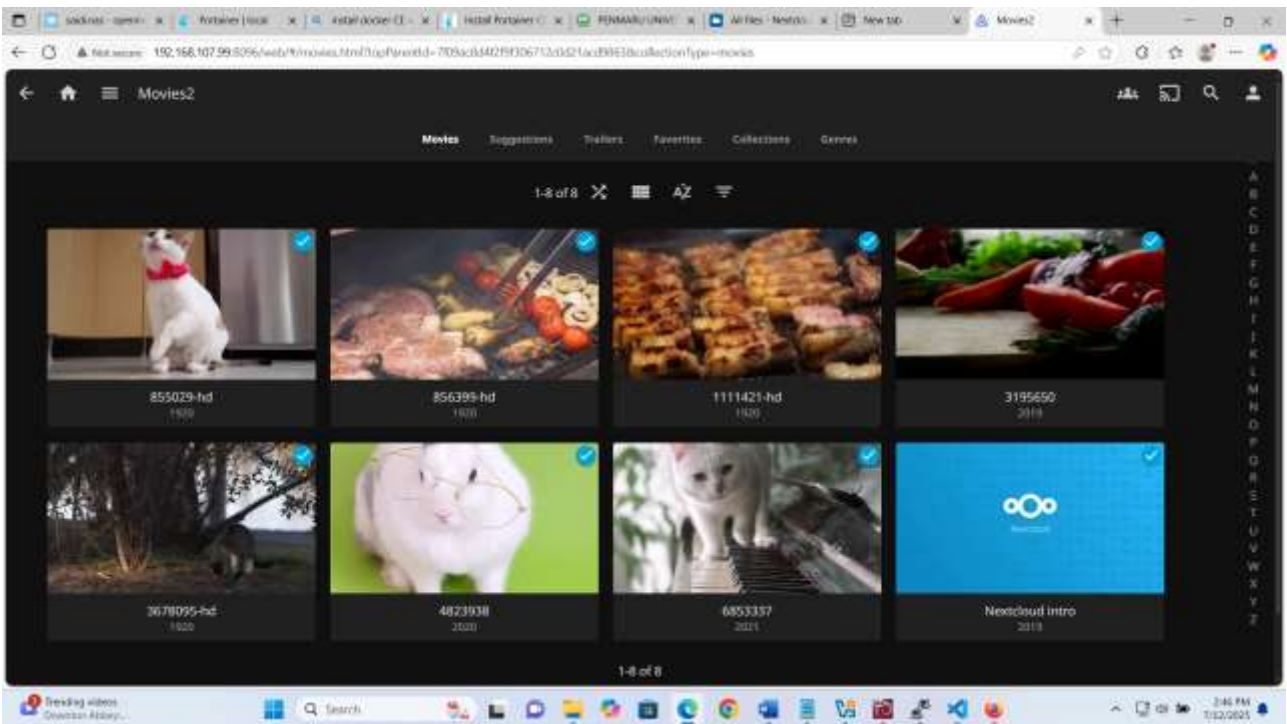
- 8096:8096

volumes:

- /path/ke/config:/config

- /path/ke/media:/media

restart: unless-stopped



Gambar 10. Tampilan Jellyfin

3.1.6 Evaluasi Deployment

Dari tahapan-tahapan dilakukan dapat disimpulkan bahwa implementas aplikasi yang digunakan sangat mudah dan juga cepat dalam melakukan penyebara apalikasi cloud computing berikut hasil deploy aplikasi yang diakukan di lingkungan teknologi yang digunakan

Table 1. Hasil Evaluasi Deploy

Aplikasi Cloud Computing	Online / Offline	Requirement Software	Waktu deployment
Nextcloud	Bisa diakses secara online dan offline	Terbaru / latest	2 menit
Jellyfin	Bisa diakses secara online dan offline	Terbaru / latest	2 menit

Pada table 3.1 di atas aplikasi yang dideploy adalah nextcloud dan jellyfin aplikasi tersebut berhasil deploy secara localhost dan memiliki aplikasi pendukung terbaru. Kemudian waktu *deploy* relatif cepat dari [dokumentasi yang dilakukan](#) tergantung kepada *internet access*.

Berikut adalah kegunaan-kegunaan utilitas software penelitian yang sudah didokumentasikan

Table 2. Fungsi Utilitas

Nama Utilitas	Web Service ?	Jenis/Peran Utama	Antarmuka Web (Web UI)	Catatan
Docker	Bukan	Platform kontainerisasi untuk menjalankan aplikasi	Tidak	Menjalankan berbagai service (termasuk web service) di dalam kontainer.
Portainer	Bukan web service melainkan <i>admin user interface</i>	Web interface untuk mengelola Docker	Ya	Admin web tool, bukan web service publik.
OpenMediaVault	Bukan web service melainkan <i>admin user interface</i>	OS NAS untuk manajemen storage dan layanan jaringan	Ya	Bisa menjalankan layanan file sharing dan server lainnya.
Nextcloud	Termasuk	<i>Web service</i> untuk penyimpanan cloud pribadi	Ya	Web-based file sync & collaboration platform mirip Google Drive.

Jellyfin	Termasuk	Web service untuk media streaming server pribadi mirip youtube	Ya	Alternatif open-source Plex, dengan UI web untuk streaming film/musik.
----------	----------	--	----	--

4. KESIMPULAN

Beberapa kesimpulan yang dapat dibuat dari tahapan-tahapan yang sudah dilakukan lengkap dengan dokumentasi yang berjumlah 52 halaman ukuran kertas A4 adalah sebagai berikut :

1. Membangun server secara mandiri mulai dari tahap awal hingga tahap deployment
2. Deployment sangat cepat tanpa ribet mengetahui versi aplikasi pendukung serta konfigurasinya
3. Beberapa web framework juga bisa berjalan dengan baik seperti Laravel

DAFTAR PUSTAKA

- [1] E. Euis and N. Heryana, "Analisis Pemanfaatan Cloud Computing Berbasis Software as a Service sebagai Media Penyimpanan Tugas Praktikum," *Syntax J. Inform.*, vol. 7, no. 1, pp. 26–32, 2018, doi: 10.35706/syji.v7i1.1187.
- [2] "View of Building E-Learning Application Using Cloud Computing with Software As A Service (SAAS) Model.pdf," 2023, *ANTIVIRUS:Jurnal Ilmiah Teknik Informatika, Binjai*. doi: <https://doi.org/10.35457/antivirus.v17i1.3172>.
- [3] N. Ruseno, S. Kurniawan, G. Santoso, and U. Teknologi Muhammadiyah Jakarta, "Penerapan Devops (Development Operations) Dalam Pengembangan Aplikasi Untuk Meningkatkan Kecepatan Delivery Software," *Jurnal*, vol. 01, no. 01, p. 2025, 2025, [Online]. Available: <https://doi.org/10.9000/jupasti.v1i1.1>
- [4] Indonet, "Konsep DevOps dan CI/CD dalam Pengembangan Aplikasi di Cloud." Accessed: Jul. 12, 2025. [Online]. Available: <https://indonet.co.id/id/konsep-devops-dan-ci-cd-dalam-pengembangan-aplikasi-di-cloud/#:~:text=Mengenal DevOps dan CI/CD&text=DevOps bertujuan untuk meningkatkan kerja,produksi jika lulus semua uji>.
- [5] Eranyacloud, "Kenali Docker dari Pengertian, Fungsi, Hingga Cara Kerjanya," eranyacloud. Accessed: Jul. 12, 2025. [Online]. Available: <https://eranyacloud.com/id/blog/docker-adalah/>
- [6] Ekobahranadinata, "Panduan Lengkap Menggunakan Docker: Apa, Mengapa, dan Bagaimana?" Accessed: Jul. 12, 2025. [Online]. Available: <https://qs.telkomuniversity.ac.id/panduan-lengkap-menggunakan-docker-apa-mengapa-dan-bagaimana/>
- [7] Z. M. Subekti and D. P. Kuswandono, "Rancang Bangun Cloud Storage Berbasis Docker Container," vol. 01, no. 01, pp. 63–71, 2025.
- [8] www.linuxsec.org, "Portainer." Accessed: Jun. 21, 2021. [Online]. Available: <https://www.linuxsec.org/2020/12/cara-install-portainer.html>
- [9] D. P. Lita, Heliyanti Susana, Martanto, Saeful Anwar, and Cep Lukman Rohmat, "Analisis Keandalan Network Attached Storage Berbasis Raspberry Pi Menggunakan Metode Client-Server," *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, vol. 5, no. 1, pp. 1–7, 2021, doi: 10.32485/kopertip.v5i1.134.
- [10] Openmediavaultteam, "What is openmediavault?," 2023. Accessed: Jul. 14, 2025. [Online]. Available: <https://www.openmediavault.org/>
- [11] "Why Nextcloud Files?," 2022. Accessed: Jul. 14, 2025. [Online]. Available: <https://nextcloud.com/files/>
- [12] Jellyfin, "Welcome to the Jellyfin Documentation." Accessed: Jul. 14, 2025. [Online]. Available: <https://jellyfin.org/docs/>