

LAPORAN PROYEK AKHIR
MEMBANGUN PRIVATE CLOUD CLUSTER DENGAN PROXMOX VE

Disusun untuk memenuhi tugas akhir mata kuliah Sistem Operasi

Dosen Pengampu: Ferdi Cahyadi, S.Kom., M.Cs



Disusun Oleh:

Kelompok 8

- | | |
|----------------------------------|------------|
| 1. Dhini Khairunnisa | 2401020016 |
| 2. Erlinda Amira Putri Sudarmono | 2401020021 |
| 3. Nikita Arzetty Siregar | 2401020030 |
| 4. Rismayanti | 2401020041 |
| 5. Ary Subakti | 2301020065 |

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO DAN INFORMATIKA
FAKULTAS TEKNIK DAN TEKNOLOGI KEMARITIMAN
UNIVERSITAS MARITIM RAJA ALI HAJI
TAHUN 2025

ABSTRAK

Proyek ini bertujuan untuk merancang dan membangun private cloud cluster berbasis Proxmox Virtual Environment dengan konfigurasi multi-node serta penyimpanan bersama menggunakan Ceph atau ZFS. Sistem yang dibangun dirancang agar mampu mendukung fitur live migration dan high availability (HA), sehingga virtual machine (VM) dapat berpindah antar node tanpa mengalami gangguan layanan serta tetap menjaga ketersediaan sistem apabila terjadi kegagalan pada salah satu node.

Berdasarkan landasan teori dan studi literatur, Proxmox VE memiliki dukungan terintegrasi dalam pembentukan cluster, mekanisme migrasi VM secara online, serta pengelolaan penyimpanan terdistribusi yang fleksibel. Pada proyek ini, implementasi sistem diuji melalui skenario live migration dan failover untuk mengevaluasi performa, stabilitas, dan keandalan cluster yang dibangun.

Melalui tahapan implementasi dan pengujian tersebut, diharapkan sistem cluster yang dihasilkan dapat beroperasi secara optimal, mendukung proses migrasi VM dengan baik, serta menyediakan dokumentasi arsitektur yang dapat dijadikan referensi dalam penerapan teknologi virtualisasi dan manajemen infrastruktur berbasis cloud.

KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga laporan final project mata kuliah Sistem Operasi dengan judul “Membangun Private Cloud Cluster dengan Proxmox VE” dapat diselesaikan sesuai dengan waktu yang telah ditentukan.

Kami mengucapkan terima kasih kepada dosen pengampu mata kuliah Sistem Operasi atas arahan dan bimbingan yang telah diberikan. Ucapan terima kasih juga kami sampaikan kepada seluruh anggota kelompok atas kerja sama, dukungan, dan kontribusi selama pelaksanaan project, serta kepada semua pihak yang turut membantu dalam penyelesaian laporan ini.

Kami menyadari bahwa laporan ini masih memiliki keterbatasan dan kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat kami harapkan sebagai bahan evaluasi dan perbaikan ke depannya. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi dalam memahami penerapan teknologi virtualisasi dan sistem cluster.

Tanjungpinang, 24 Desember 2025

Kelompok 8

DAFTAR ISI

ABSTRAK.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Project.....	2
BAB II LANDASAN TEORI.....	4
2.1 Proxmox Virtual Environment (Proxmox VE)	4
2.2 Cluster dan High Availability (HA)	4
2.3 Storage Bersama: ZFS	5
2.4 Live Migration.....	6
2.5 High Availability (HA) dan Failover.....	6
BAB III PERANCANGAN ARSITEKTUR	8

3.1 Gambaran Umum Sistem	8
3.2 Desain Arsitektur Cluster	8
3.3 Topologi Sistem / Topologi Jaringan	9
3.4 Desain Virtualisasi.....	11
3.5 Perancangan Storage.....	11
3.6 Desain Jaringan dan Rancangan Layanan Sistem	11
3.7 Kesesuaian Arsitektur dengan Tujuan Project	12
BAB IV IMPLEMENTASI TEKNIS.....	13
4.1 Lingkungan Implementasi	13
4.2 Instalasi dan Konfigurasi Proxmox VE.....	14
4.3 Pembentukan Cluster	16
4.3 Konfigurasi Storage	17
4.3.1 Konfigurasi ZFS Storage	17
4.3.2 Konfigurasi Tipe Konten Storage	18
4.3.3 Konfigurasi Shared Storage NFS	19
4.3.4 Manajemen Storage	20
4.4 Implementasi LXC Container	22

4.4.1 Konfigurasi LXC Container.....	22
4.5 Konfigurasi Jaringan	23
4.6 Implementasi Virtual Machine	24
4.7 Implementasi Live Migration.....	25
4.8 Implementasi High Availability (HA)	26
BAB V PENGUJIAN DAN ANALISIS	28
5.1 Analisis Konektivitas Jaringan antar Node dan VM	28
5.2 Analisis Mendalam Live Migration (Manual Migration)	28
5.3 Analisis Mekanisme Failover dan High Availability (HA)	30
BAB VI PENUTUP.....	32
6.1 Kesimpulan.....	32
6.2 Saran.....	32
DAFTAR PUSTAKA.....	34
LAMPIRAN.....	35

DAFTAR GAMBAR

Gambar 1 Diagram Arsitektur	9
Gambar 2 Topologi Jaringan	10
Gambar 3 Proses Instalasi Pada Salah Satu Node.....	14
Gambar 4 Dashboard Node 3.....	15
Gambar 5 Dashboard Node 2.....	15
Gambar 6 Tampilan status cluster Proxmox VE dengan tiga node aktif	16
Gambar 7 Proses pembuatan ZFS Storage Pool	18
Gambar 8 Tampilan daftar storage dan tipe konten pada Proxmox VE.....	19
Gambar 9 Konfigurasi Shared Storage NFS pada Proxmox VE	20
Gambar 10 Manajemen Storage.....	21
Gambar 11 Konfigurasi Pembuatan LXC Container pada Proxmox VE	23
Gambar 12 Pembuatan Virtual Machine	24
Gambar 13 Proses Live Migration VM Antar-Node pada Cluster Proxmox VE	25
Gambar 14 Migrasi Sukses	26
Gambar 15 Analisis Migrasi Sukses	29

Gambar 16 After-Down-Node3	30
Gambar 17 Before-Down-Node3.....	30

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi mendorong berbagai organisasi, institusi pendidikan, dan perusahaan untuk beralih dari penggunaan server fisik tunggal menuju infrastruktur berbasis virtualisasi dan cloud. Teknologi virtualisasi memberikan berbagai keuntungan, seperti pemanfaatan sumber daya yang lebih efisien, kemudahan dalam pengelolaan sistem, serta fleksibilitas dalam penyediaan layanan. Dengan virtualisasi, beberapa sistem operasi dapat dijalankan secara terisolasi pada satu perangkat keras, sehingga proses deployment, migrasi, dan konsolidasi server menjadi lebih mudah dan efektif.

Penerapan virtualisasi yang dikombinasikan dengan arsitektur cluster memungkinkan organisasi membangun infrastruktur yang lebih scalable dan tahan terhadap gangguan. Dukungan storage bersama serta mekanisme clustering memungkinkan terjadinya redundansi dan distribusi beban kerja. Berbagai studi menunjukkan bahwa penggunaan cluster virtualisasi dengan penyimpanan terdistribusi dan sistem failover mampu meningkatkan ketersediaan layanan (availability) serta mengurangi downtime ketika terjadi kegagalan pada node atau sistem penyimpanan.

Proxmox VE merupakan salah satu platform virtualisasi open-source yang banyak digunakan karena mendukung virtual machine melalui KVM dan container melalui LXC. Platform ini menyediakan fitur manajemen terpusat, dukungan cluster, pengelolaan storage yang fleksibel baik lokal maupun shared, kemampuan live migration, serta opsi High Availability (HA). Dengan memanfaatkan Proxmox VE bersama teknologi storage terdistribusi seperti Ceph atau ZFS/shared storage, pengguna termasuk mahasiswa dapat membangun private cloud cluster dengan biaya

yang relatif terjangkau, namun tetap memiliki fitur dan performa yang mendekati infrastruktur cloud profesional.

Melalui proyek ini, mahasiswa dapat memperoleh pengalaman dalam merancang dan mengelola infrastruktur virtual dan terdistribusi, memahami kompromi antara kesederhanaan sistem dan kebutuhan redundansi serta ketersediaan tinggi, mengevaluasi mekanisme live migration, failover, dan high availability berdasarkan hasil pengujian nyata, serta mendokumentasikan arsitektur, konfigurasi, dan hasil pengujian secara sistematis.

1.2 Rumusan Masalah

1. Bagaimana perancangan arsitektur private cloud cluster yang diterapkan menggunakan Proxmox VE?
2. Bagaimana proses instalasi serta konfigurasi Proxmox VE pada lingkungan cluster multi-node?
3. Bagaimana tahapan pembangunan dan integrasi sistem storage bersama menggunakan Ceph atau ZFS dalam cluster?
4. Bagaimana metode pengujian mekanisme live migration antar node pada cluster yang telah dibangun?
5. Bagaimana cara memastikan stabilitas cluster serta penerapan High Availability (HA) melalui mekanisme failover?

1.3 Tujuan Project

1. Untuk mempelajari perancangan arsitektur private cloud cluster yang stabil dan mampu dikembangkan (scalable) dengan memanfaatkan Proxmox VE.
2. Untuk memahami tahapan instalasi serta konfigurasi Proxmox VE pada lingkungan multi-node agar cluster dapat beroperasi secara optimal.

3. Untuk mempelajari proses konfigurasi storage bersama menggunakan Ceph atau ZFS sehingga virtual machine dapat diakses oleh seluruh node dalam cluster.
4. Untuk memahami mekanisme live migration virtual machine antar node tanpa menimbulkan downtime, sehingga ketersediaan layanan tetap terjaga.
5. Untuk mengevaluasi penerapan mekanisme failover dan High Availability (HA) ketika terjadi kegagalan pada salah satu node dalam cluster.

BAB II

LANDASAN TEORI

2.1 Proxmox Virtual Environment (Proxmox VE)

Proxmox VE (Virtual Environment) merupakan platform virtualisasi open-source berbasis sistem operasi Debian yang mendukung teknologi KVM untuk menjalankan mesin virtual serta LXC untuk pengelolaan container. Platform ini menyediakan antarmuka web terintegrasi yang memudahkan administrator dalam mengelola virtual machine, container, penyimpanan, jaringan, proses pencadangan, hingga manajemen cluster secara terpusat.

Beberapa keunggulan utama Proxmox VE antara lain:

1. Mendukung pembentukan cluster multi-node, sehingga beberapa server fisik dapat dikelola sebagai satu kesatuan sistem.
2. Menyediakan fitur live migration yang memungkinkan perpindahan virtual machine antar node tanpa menyebabkan gangguan layanan.
3. Mendukung pengelolaan storage yang fleksibel, baik penyimpanan lokal maupun terdistribusi seperti Ceph, ZFS, NFS, dan iSCSI.
4. Memiliki dukungan High Availability (HA) untuk memastikan virtual machine tetap berjalan ketika terjadi kegagalan pada salah satu node.

2.2 Cluster dan High Availability (HA)

Cluster merupakan sekumpulan beberapa server (node) yang saling terhubung dan bekerja secara bersama-sama sehingga dapat dikelola sebagai satu kesatuan sistem. Dengan konsep cluster, sumber daya seperti CPU, RAM, dan storage dapat dibagi serta

dimanfaatkan secara efisien, sekaligus memungkinkan pengelolaan sistem secara terpusat.

High Availability (HA) adalah kemampuan suatu sistem untuk tetap menjaga ketersediaan layanan meskipun terjadi gangguan atau kegagalan pada salah satu node. Pada Proxmox VE, penerapan HA dilakukan melalui integrasi antara cluster, penggunaan shared storage, serta pengelola HA (HA manager). Kombinasi tersebut memungkinkan virtual machine untuk dipindahkan atau dijalankan kembali secara otomatis ketika salah satu node mengalami kegagalan, sehingga layanan tetap berjalan.

2.3 Storage Bersama: ZFS

ZFS (Zettabyte File System) merupakan sistem file dan manajemen storage yang dirancang untuk menyediakan keandalan, integritas data, serta kemudahan pengelolaan penyimpanan. ZFS menggabungkan fungsi file system dan volume manager dalam satu sistem, sehingga mampu mengelola storage secara lebih efisien dan aman.

Dalam lingkungan virtualisasi seperti pada Proxmox VE, ZFS sering digunakan sebagai storage bersama (shared storage) karena mendukung fitur-fitur penting seperti pooling storage, snapshot, dan data integrity checking. Dengan mekanisme copy-on-write, ZFS mampu mencegah kerusakan data serta menjaga konsistensi data pada saat terjadi gangguan sistem.

Penggunaan ZFS sebagai storage bersama memungkinkan virtual machine (VM) disimpan pada satu pool storage yang dapat diakses oleh beberapa node dalam cluster. Hal ini sangat mendukung penerapan live migration dan High Availability (HA), karena VM dapat dipindahkan atau dijalankan kembali pada node lain tanpa

kehilangan data. Selain itu, ZFS juga menyediakan fitur redundancy melalui konfigurasi RAID-Z, sehingga meningkatkan keandalan sistem penyimpanan.

2.4 Live Migration

Live migration merupakan mekanisme pemindahan virtual machine (VM) dari satu node ke node lain dalam sebuah cluster tanpa menghentikan layanan yang sedang berjalan. Dengan fitur ini, VM tetap aktif selama proses migrasi berlangsung sehingga tidak menimbulkan downtime dan gangguan terhadap pengguna.

Pada lingkungan cluster berbasis Proxmox VE, live migration didukung secara terintegrasi dengan memanfaatkan jaringan cluster dan storage bersama. Keberadaan shared storage memungkinkan data VM tetap konsisten dan dapat diakses oleh node tujuan, sehingga proses perpindahan VM dapat dilakukan secara cepat dan aman.

Live migration umumnya digunakan untuk keperluan pemeliharaan sistem, seperti pembaruan perangkat lunak, perbaikan node, atau penyeimbangan beban kerja antar node. Selain itu, fitur ini juga berperan penting dalam mendukung High Availability (HA), karena memungkinkan VM dipindahkan secara dinamis ketika terdeteksi adanya gangguan pada salah satu node.

2.5 High Availability (HA) dan Failover

High Availability (HA) merupakan konsep yang bertujuan untuk menjaga ketersediaan layanan agar tetap berjalan meskipun terjadi gangguan atau kegagalan pada salah satu komponen sistem. Dalam lingkungan virtualisasi, HA sangat penting untuk meminimalkan downtime dan memastikan layanan tetap dapat diakses oleh pengguna.

Failover adalah mekanisme yang memungkinkan sistem secara otomatis mengalihkan layanan atau virtual machine (VM) ke node lain ketika node utama mengalami kegagalan. Dengan adanya failover, VM yang terdampak dapat dijalankan kembali pada node yang masih aktif tanpa memerlukan intervensi manual dari administrator.

Pada cluster berbasis Proxmox VE, penerapan HA dilakukan melalui integrasi antara cluster, shared storage, serta HA manager. Sistem secara terus-menerus memantau kondisi setiap node dalam cluster. Apabila terdeteksi adanya node yang down atau tidak responsif, HA manager akan mengeksekusi proses failover dengan memindahkan atau menjalankan ulang VM pada node lain yang tersedia.

BAB III

PERANCANGAN ARSITEKTUR

3.1 Gambaran Umum Sistem

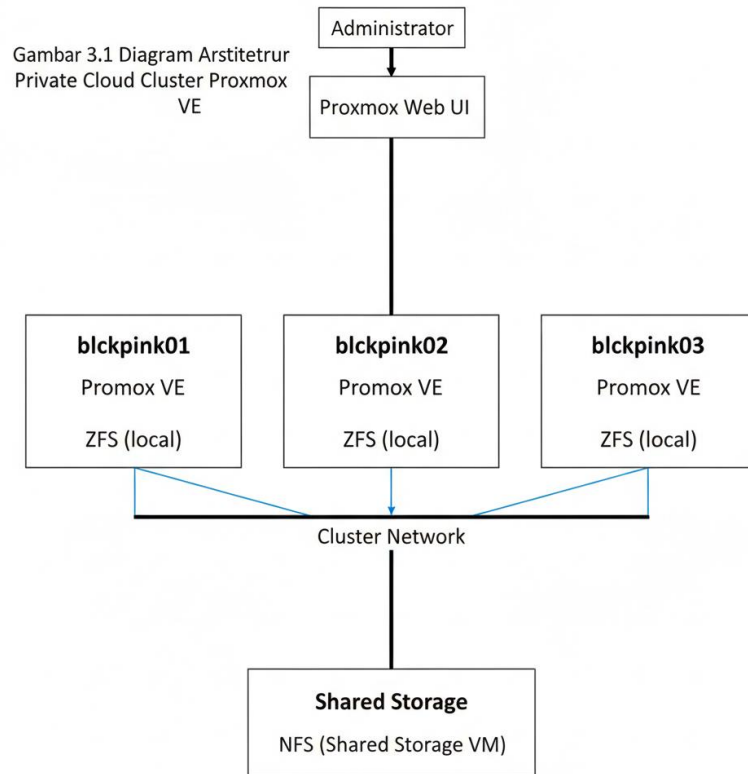
Pada proyek ini dirancang sebuah private cloud cluster berbasis Proxmox VE yang terdiri dari tiga node server, yaitu blckpink01, blckpink02, dan blckpink03. Ketiga node tersebut tergabung dalam satu cluster sehingga dapat berbagi sumber daya dan dikelola sebagai satu sistem terpusat.

Arsitektur sistem dirancang untuk mendukung virtualisasi server, penggunaan storage bersama, serta mekanisme live migration dan failover. Dengan desain ini, sistem diharapkan mampu menyediakan layanan yang stabil, fleksibel, dan tetap tersedia meskipun terjadi kegagalan pada salah satu node.

3.2 Desain Arsitektur Cluster

Cluster Proxmox VE dirancang menggunakan tiga node yang memiliki peran setara sebagai hypervisor. Seluruh node saling terhubung melalui jaringan cluster sehingga dapat berkomunikasi dan berbagi informasi status sistem.

Dalam arsitektur ini, virtual machine dan container dapat dijalankan pada salah satu node dan dipindahkan ke node lain tanpa menghentikan layanan. Penggunaan multi-node cluster memungkinkan peningkatan ketersediaan sistem serta mendukung penerapan High Availability (HA).



Gambar 1 Diagram Arsitektur

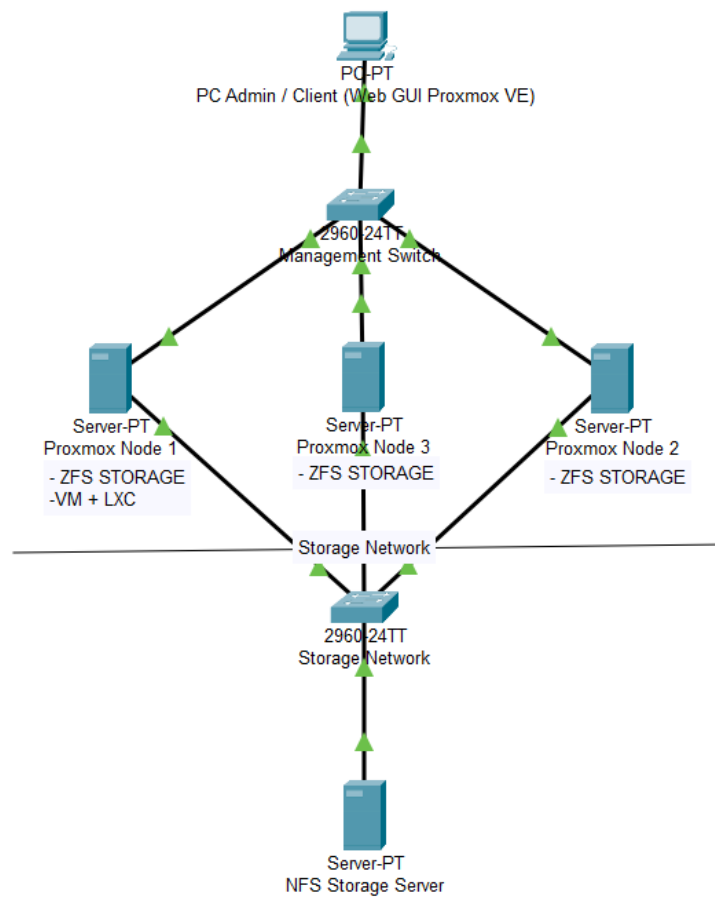
Diagram menunjukkan tiga node Proxmox VE yang tergabung dalam satu cluster dan terhubung melalui jaringan cluster. Seluruh node mengakses shared storage berbasis NFS sebagai lokasi penyimpanan disk virtual machine, sedangkan ZFS digunakan sebagai local storage pada masing-masing node.

3.3 Topologi Sistem / Topologi Jaringan

Topologi jaringan Proxmox Virtual Environment (VE) dengan pendekatan enterprise, yang terdiri dari 3 node Proxmox, pemisahan Management Network dan Storage Network, serta penggunaan ZFS dan NFS Storage.

Topologi ini dibuat menggunakan Cisco Packet Tracer dan disajikan tanpa konfigurasi jaringan. Topologi dirancang untuk mendukung:

- Cluster Proxmox VE 3 Node
- High Availability (HA)
- Pemisahan trafik manajemen dan penyimpanan
- Virtual Machine dan LXC Container



Gambar 2 Topologi Jaringan

3.4 Desain Virtualisasi

Lingkungan virtualisasi pada sistem ini mendukung dua jenis teknologi, yaitu Virtual Machine (QEMU) dan LXC Container. Penggunaan VM dan container bertujuan untuk menguji fleksibilitas Proxmox VE dalam menangani berbagai jenis workload.

Dengan kombinasi VM dan container, sistem dapat digunakan untuk menjalankan layanan dengan kebutuhan sumber daya yang berbeda-beda secara efisien.

3.5 Perancangan Storage

Sistem penyimpanan dirancang menggunakan dua jenis storage, yaitu shared storage dan local storage. Shared storage berbasis NFS digunakan sebagai lokasi penyimpanan disk virtual machine sehingga dapat diakses oleh seluruh node dalam cluster. Konfigurasi ini mendukung proses live migration dan failover tanpa kehilangan data.

Selain itu, setiap node juga memiliki local storage berbasis ZFS yang digunakan untuk kebutuhan sistem dan pengelolaan lokal. Kombinasi shared storage dan local storage ini dirancang untuk meningkatkan fleksibilitas serta keandalan sistem.

3.6 Desain Jaringan dan Rancangan Layanan Sistem

Dari sisi jaringan, seluruh node menggunakan bridge jaringan vmbr0. Konfigurasi ini memastikan bahwa virtual machine dan container tetap berada pada jaringan yang sama meskipun berpindah node selama proses live migration.

Desain jaringan ini mendukung komunikasi antar node, akses storage bersama, serta konektivitas layanan yang berjalan di dalam VM dan container.

Layanan yang dirancang pada sistem ini meliputi:

1. Layanan virtualisasi virtual machine (VM)
2. Layanan container (LXC)
3. Layanan cluster management
4. Layanan live migration
5. Layanan failover dan High Availability
6. Layanan manajemen terpusat melalui web interface

Seluruh layanan tersebut dikelola secara terpusat melalui Proxmox VE sehingga memudahkan administrator dalam pengelolaan sistem.

3.7 Kesesuaian Arsitektur dengan Tujuan Project

Arsitektur sistem yang dirancang telah sesuai dengan tujuan project, yaitu memahami konsep cluster, live migration, dan High Availability. Penggunaan tiga node cluster mendukung pembagian resource dan ketersediaan layanan. Shared storage berbasis NFS memungkinkan live migration dan failover dilakukan tanpa kehilangan data, sementara penggunaan VM dan container menunjukkan fleksibilitas sistem terhadap berbagai workload.

BAB IV

IMPLEMENTASI TEKNIS

4.1 Lingkungan Implementasi

Implementasi sistem private cloud dilakukan pada lingkungan cluster berbasis Proxmox VE yang terdiri dari tiga node server, yaitu blckpink01, blckpink02, dan blckpink03. Ketiga node tersebut tergabung dalam satu cluster sehingga dapat saling berkomunikasi dan berbagi sumber daya seperti CPU, RAM, dan storage.

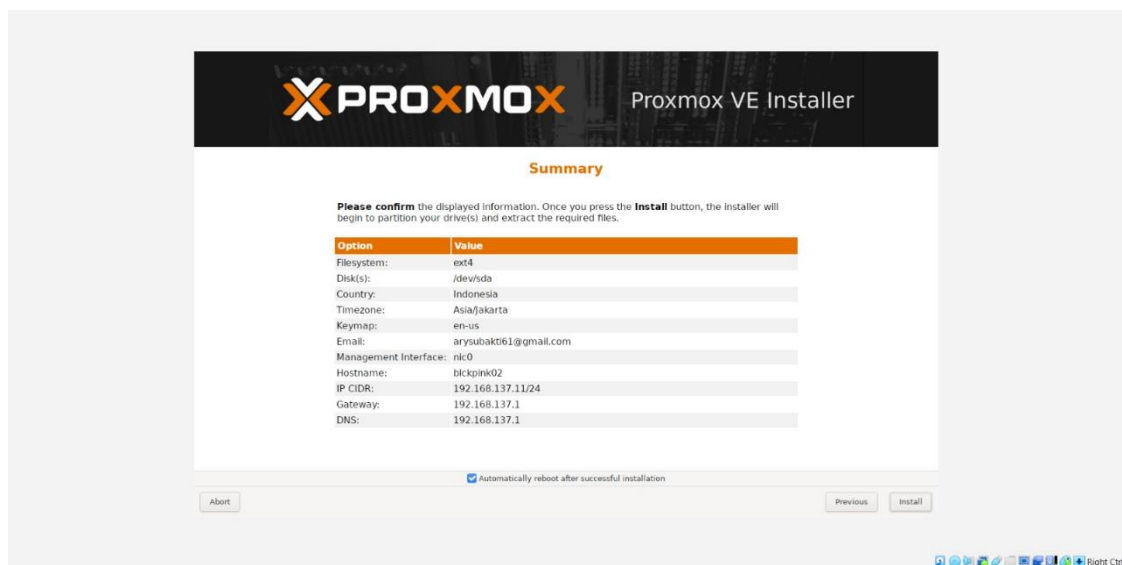
Teknologi virtualisasi yang digunakan dalam sistem ini meliputi Virtual Machine (QEMU/KVM) dan LXC Container. Penggunaan dua jenis virtualisasi ini bertujuan untuk menguji fleksibilitas Proxmox VE dalam menangani berbagai jenis workload dengan karakteristik yang berbeda.

Dari sisi penyimpanan, sistem menggunakan shared storage berbasis NFS sebagai lokasi penyimpanan disk virtual machine. Penggunaan shared storage memungkinkan virtual machine diakses oleh seluruh node dalam cluster, sehingga mendukung proses live migration dan failover tanpa kehilangan data. Selain itu, setiap node juga dilengkapi dengan local storage berbasis ZFS yang digunakan untuk kebutuhan sistem dan pengelolaan penyimpanan lokal.

Pada aspek jaringan, seluruh node dikonfigurasi menggunakan network bridge vmbr0. Konfigurasi ini memastikan virtual machine dan container tetap berada pada jaringan yang sama meskipun berpindah node selama proses live migration.

4.2 Instalasi dan Konfigurasi Proxmox VE

Tahap implementasi diawali dengan instalasi Proxmox VE pada setiap node server, yaitu blkcpink01, blkcpink02, dan blkcpink03. Instalasi dilakukan secara bare metal menggunakan installer resmi Proxmox VE. Pada tahap ini dilakukan konfigurasi dasar sistem yang meliputi pengaturan hostname, alamat IP statis, gateway, DNS, zona waktu, serta media penyimpanan sistem menggunakan local storage berbasis ZFS.

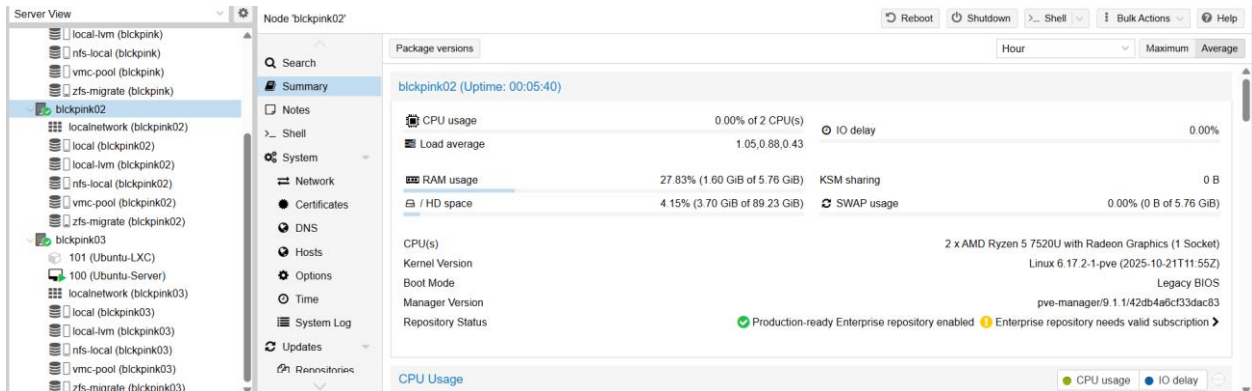


Gambar 3 Proses Instalasi Pada Salah Satu Node

Setelah proses instalasi selesai, masing-masing node dapat diakses melalui antarmuka web Proxmox VE menggunakan browser. Antarmuka web ini berfungsi sebagai pusat manajemen untuk konfigurasi lanjutan, pemantauan kondisi node, serta pengelolaan virtual machine dan container.

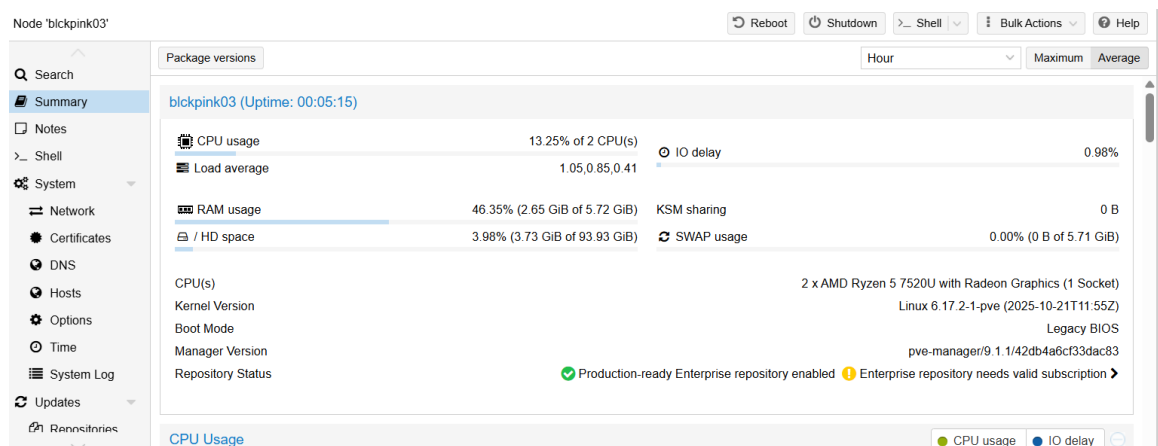
Dashboard pada Node 2 menampilkan kondisi sistem hypervisor Proxmox VE dalam keadaan aktif dan stabil. Informasi penggunaan CPU, memori, serta status node menunjukkan bahwa Node 2 telah berhasil diinstal dan siap digunakan sebagai bagian

dari cluster. Node ini berperan sebagai salah satu host untuk menjalankan layanan Virtual Machine dan LXC Container.



Gambar 5 Dashboard Node 2

Dashboard pada Node 3 menunjukkan bahwa hypervisor Proxmox VE telah berjalan dengan baik dan terhubung dalam lingkungan cluster. Status node dalam kondisi online menandakan bahwa Node 3 siap mendukung proses implementasi layanan virtualisasi, termasuk pemindahan layanan antar node dan mekanisme high availability.

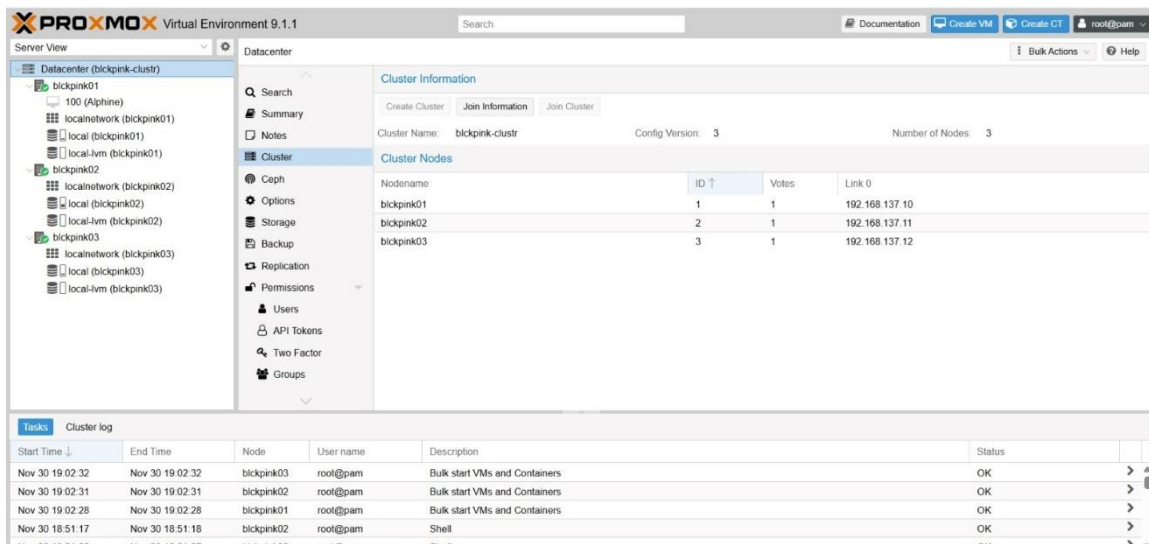


Gambar 4 Dashboard Node 3

4.3 Pembentukan Cluster

Langkah selanjutnya adalah pembentukan cluster Proxmox VE. Cluster dibuat dengan menentukan nama cluster blckpink-clustr serta jaringan cluster yang digunakan untuk komunikasi antar-node. Setelah cluster berhasil dibuat, node lain kemudian ditambahkan (join) ke dalam cluster sehingga seluruh node dapat dikelola sebagai satu sistem terpadu.

Berdasarkan hasil konfigurasi, cluster Proxmox VE berhasil dibentuk dengan total tiga node aktif. Setiap node memiliki satu suara (vote) sehingga mekanisme quorum dapat tercapai menggunakan mayoritas node. Dengan konfigurasi ini, cluster berada dalam kondisi stabil dan siap mendukung fitur live migration serta high availability.



Gambar 6 Tampilan status cluster Proxmox VE dengan tiga node aktif

Tabel 4.1 Konfigurasi Alamat IP Node Cluster

Node	Alamat IP
blckpink01	192.168.137.10
blckpink02	192.168.137.11
blckpink03	192.168.137.12

4.3 Konfigurasi Storage

Pada tahap ini dilakukan konfigurasi media penyimpanan (storage) yang digunakan pada sistem Proxmox VE. Storage berfungsi sebagai lokasi penyimpanan disk virtual, container, file ISO, serta data cadangan (backup) sistem.

4.3.1 Konfigurasi ZFS Storage

Sistem menggunakan ZFS (Zettabyte File System) sebagai storage lokal pada node Proxmox. ZFS dikonfigurasi dengan membuat sebuah storage pool bernama blckpink-storage-pool yang menggunakan satu perangkat disk (/dev/sdb) dengan skema single disk.

Pengaturan kompresi diaktifkan untuk meningkatkan efisiensi penyimpanan, serta nilai ashift disesuaikan untuk kompatibilitas sektor disk. ZFS dipilih karena memiliki keunggulan dalam hal keandalan data, manajemen storage yang fleksibel, serta performa yang stabil untuk kebutuhan virtualisasi.

Create: ZFS

Name:

Add Storage: ☒

RAID Level:

Compression:

ashift:

<input checked="" type="checkbox"/>	Device ↑	Model	Serial	Size	Order
<input checked="" type="checkbox"/>	/dev/sdb	VBOX_HARDDISK	VB1b274aa3-d9f16b31	405.47 GB	

Note: ZFS is not compatible with disks backed by a hardware RAID controller. For details see [the reference documentation](#).

Gambar 7 Proses pembuatan ZFS Storage Pool

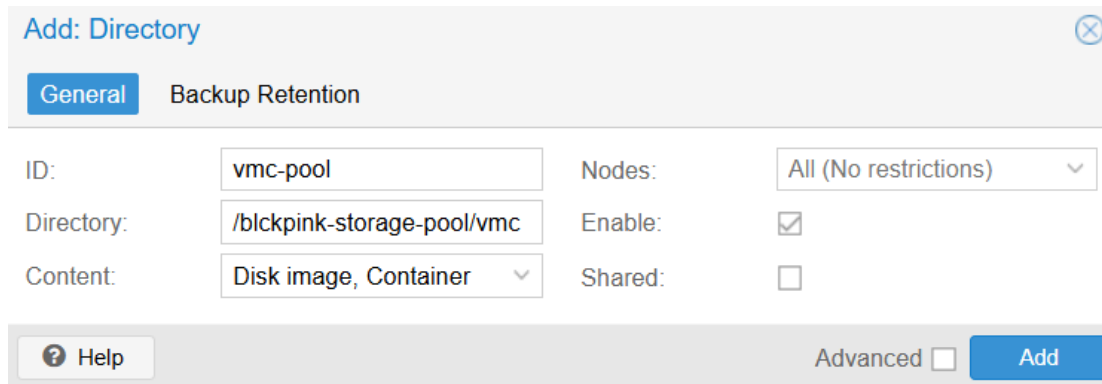
4.3.2 Konfigurasi Tipe Konten Storage

Setelah storage berhasil dibuat, Proxmox VE menampilkan beberapa tipe konten yang dapat dikelola pada storage tersebut. Tipe konten ini menentukan fungsi storage dalam sistem virtualisasi.

Adapun tipe konten storage yang tersedia meliputi:

- Disk Image, digunakan untuk menyimpan disk virtual mesin virtual (VM)
- Container, digunakan untuk menyimpan data LXC Container
- ISO Image, digunakan sebagai media instalasi sistem operasi
- Backup, digunakan untuk menyimpan hasil pencadangan VM dan container

Dengan adanya pembagian tipe konten ini, administrator dapat mengelola data virtualisasi secara terstruktur dan memudahkan proses pemeliharaan sistem, termasuk backup dan pemulihan data.



The screenshot shows the 'Add: Directory' window in Proxmox VE. It has a title bar with a close button. Below the title bar are two tabs: 'General' and 'Backup Retention'. The 'General' tab is active and contains the following fields:

- ID: vmc-pool
- Nodes: All (No restrictions) (dropdown menu)
- Directory: /blckpink-storage-pool/vmc
- Content: Disk image, Container (dropdown menu)
- Enable: ☒
- Shared: ☐

At the bottom of the window, there is a 'Help' button with a question mark icon, an 'Advanced' checkbox, and an 'Add' button.

Gambar 8 Tampilan daftar storage dan tipe konten pada Proxmox VE

4.3.3 Konfigurasi Shared Storage NFS

Selain storage lokal berbasis ZFS, sistem ini juga menggunakan shared storage berbasis NFS (Network File System). Shared storage ini berfungsi sebagai media penyimpanan bersama antar node dalam cluster Proxmox VE, sehingga mendukung fitur live migration dan failover tanpa kehilangan data.

Konfigurasi NFS dilakukan melalui menu Datacenter → Storage → Add → NFS pada Proxmox Web UI. Pada proses ini, administrator mengisi beberapa parameter utama, yaitu:

1. ID, Digunakan sebagai nama identitas storage NFS di Proxmox.
2. Server, Alamat IP server NFS yang menyediakan shared storage.
3. Export, Direktori yang dibagikan oleh server NFS.
4. Content, Jenis data yang disimpan, seperti Disk Image, Container, dan Backup.

Setelah konfigurasi disimpan, storage NFS akan muncul pada daftar storage di seluruh node cluster. Dengan demikian, VM dan container dapat berpindah antar node tanpa memindahkan data secara manual.

The screenshot shows the 'Add: NFS' configuration interface in Proxmox VE. The 'General' tab is active, displaying the following settings:

- ID:** nfs-local
- Nodes:** blkpink
- Server:** 192.168.1.10
- Export:** /srv/nfs/proxmox
- Content:** Disk image, ISO image
- Enable:** ☒

At the bottom of the window, there is a 'Help' button, an 'Advanced' checkbox, and an 'Add' button.

Gambar 9 Konfigurasi Shared Storage NFS pada Proxmox VE

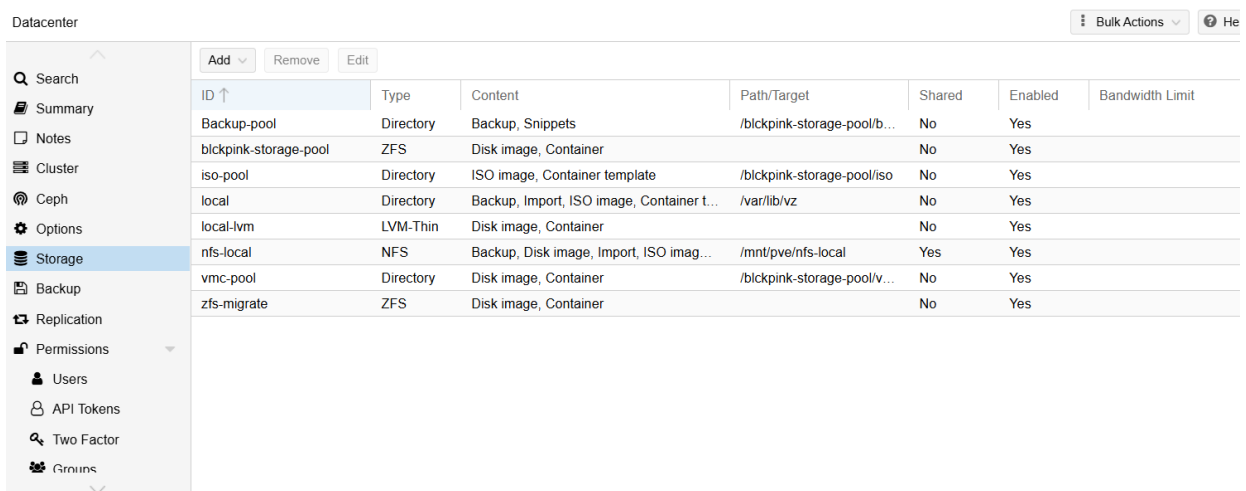
4.3.4 Manajemen Storage

Dalam membangun Private Cloud Cluster, manajemen penyimpanan (storage) menjadi komponen paling vital. Proxmox VE memungkinkan penggunaan berbagai jenis storage backend untuk kebutuhan yang berbeda. Pada proyek ini, kami mengonfigurasi tiga jenis storage utama:

1. ZFS (Local): Digunakan pada masing-masing node untuk memberikan performa I/O yang tinggi, fitur data integrity, dan kemampuan snapshot yang cepat pada level lokal.
2. LVM-Thin: Digunakan untuk alokasi disk virtual yang efisien, di mana ruang penyimpanan hanya akan terpakai sebesar data yang benar-benar ditulis oleh VM.
3. Shared Storage (NFS): Ini adalah bagian terpenting untuk fitur High Availability dan Live Migration. Dengan menggunakan protokol NFS, file ISO dan disk image VM disimpan di satu server terpusat yang dapat diakses

secara bersamaan oleh semua node dalam cluster. Tanpa shared storage, proses migrasi VM akan memakan waktu lama karena harus menyalin seluruh isi disk antar node.

Berikut adalah ringkasan status storage yang telah berhasil dikonfigurasi dan terdeteksi aktif pada seluruh node cluster:



ID ↑	Type	Content	Path/Target	Shared	Enabled	Bandwidth Limit
Backup-pool	Directory	Backup, Snippets	/blckpink-storage-pool/b...	No	Yes	
blckpink-storage-pool	ZFS	Disk image, Container		No	Yes	
iso-pool	Directory	ISO image, Container template	/blckpink-storage-pool/iso	No	Yes	
local	Directory	Backup, Import, ISO image, Container t...	/var/lib/vz	No	Yes	
local-lvm	LVM-Thin	Disk image, Container		No	Yes	
nfs-local	NFS	Backup, Disk image, Import, ISO imag...	/mnt/pve/nfs-local	Yes	Yes	
vmc-pool	Directory	Disk image, Container	/blckpink-storage-pool/v...	No	Yes	
zfs-migrate	ZFS	Disk image, Container		No	Yes	

Gambar 10 Manajemen Storage

Gambar tersebut menunjukkan status akhir dari seluruh media penyimpanan yang terdaftar pada Datacenter Proxmox. Terlihat bahwa sistem memiliki beberapa jenis storage seperti ZFS (blckpink-storage-pool), LVM-Thin, dan yang paling krusial untuk fitur Live Migration adalah nfs-local yang berstatus Shared: Yes. Seluruh storage dalam kondisi aktif (Enabled: Yes) dan siap digunakan.

4.4 Implementasi LXC Container

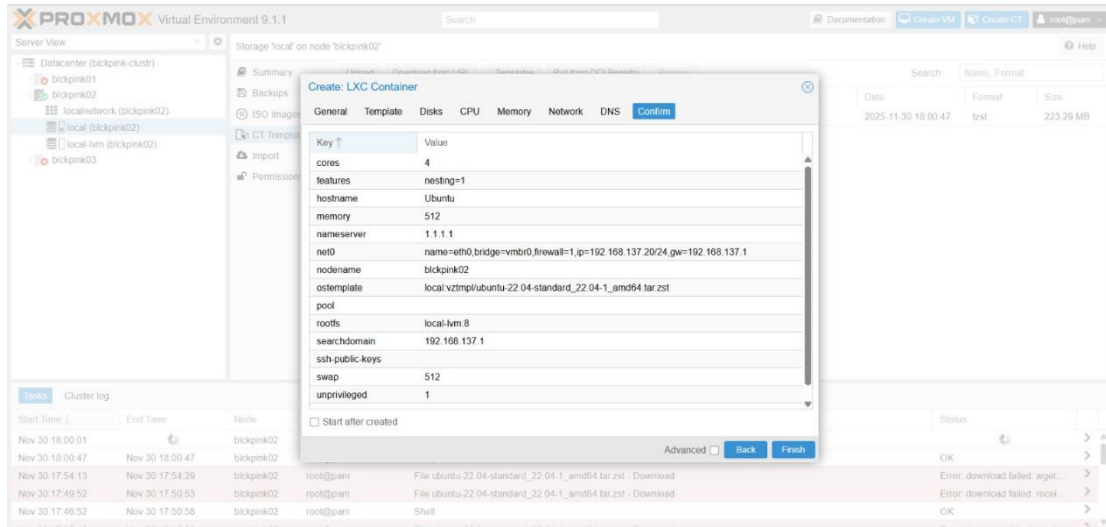
Pada tahap ini dilakukan implementasi layanan virtualisasi menggunakan LXC (Linux Container) pada cluster Proxmox VE. LXC dipilih karena bersifat ringan, efisien, dan cocok digunakan untuk pengujian layanan pada lingkungan cluster.

Proses pembuatan container dilakukan melalui Proxmox Web Interface dengan memilih menu Create CT. Container dibuat pada salah satu node cluster dengan menggunakan template sistem operasi Ubuntu 22.04.

4.4.1 Konfigurasi LXC Container

Konfigurasi utama yang digunakan pada pembuatan container antara lain:

- Hostname: Ubuntu
- Jumlah CPU: 4 core
- Memory: 512 MB
- Swap: 512 MB
- Template OS: Ubuntu 22.04
- Storage: local-lvm
- Network: Bridge vmbr0
- IP Address: 192.168.137.20/24
- Gateway: 192.168.137.1
- DNS: 1.1.1.1
- Mode: Unprivileged Container



Gambar 11 Konfigurasi Pembuatan LXC Container pada Proxmox VE

4.5 Konfigurasi Jaringan

Konfigurasi jaringan pada sistem cluster dilakukan pada setiap node Proxmox VE menggunakan metode bridge network. Bridge jaringan yang digunakan adalah `vmbr0`, yang berfungsi sebagai penghubung antara jaringan fisik host dengan Virtual Machine dan LXC Container.

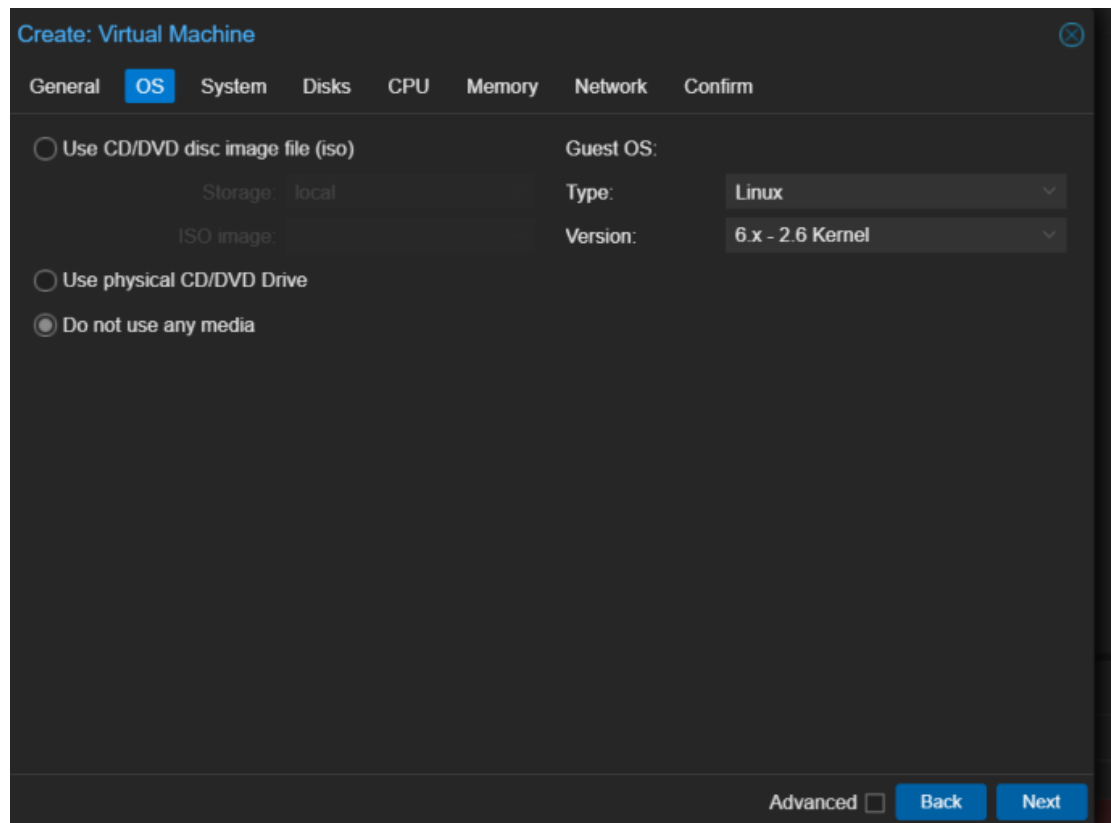
Bridge `vmbr0` dikonfigurasi agar terhubung langsung ke network interface fisik server sehingga memungkinkan VM dan container mengakses jaringan lokal serta berkomunikasi antar node dalam cluster.

Seluruh node dalam cluster menggunakan konfigurasi bridge jaringan `vmbr0` dengan pengaturan subnet dan gateway yang sama. Konsistensi konfigurasi ini memastikan bahwa VM dan container tetap berada pada satu segmen jaringan meskipun dijalankan pada node yang berbeda.

Konfigurasi ini menjadi dasar utama dalam mendukung fitur live migration dan high availability, karena perpindahan layanan dapat dilakukan tanpa gangguan konektivitas jaringan.

4.6 Implementasi Virtual Machine

Pada tahap ini dilakukan pembuatan sebuah Virtual Machine (VM) yang digunakan sebagai objek uji live migration antar-node cluster.



Gambar 12 Pembuatan Virtual Machine

4.7 Implementasi Live Migration

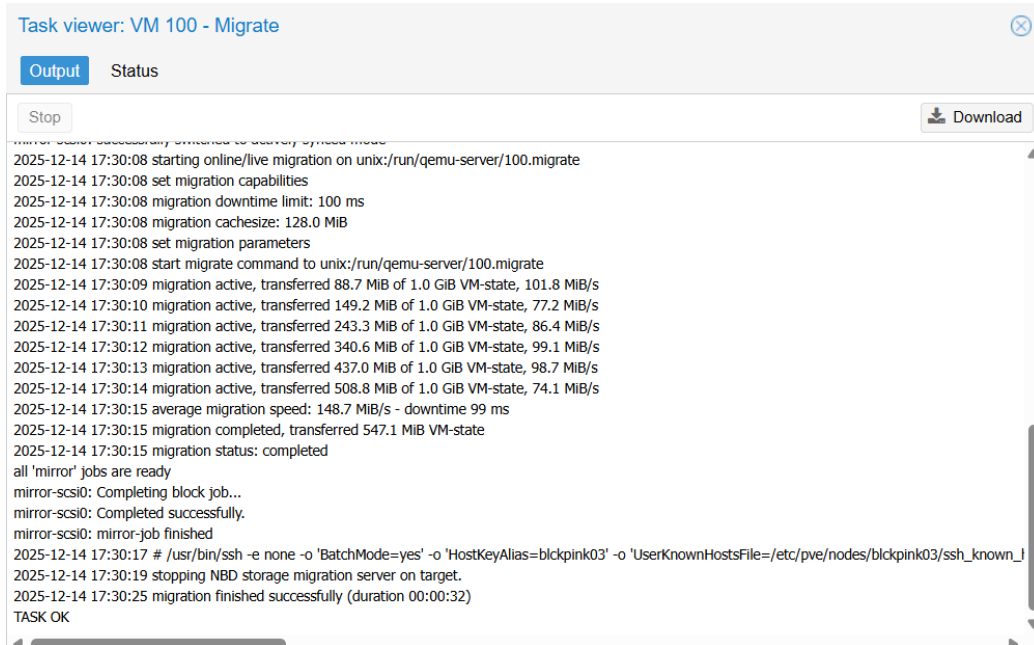
Fitur live migration diimplementasikan pada sistem cluster Proxmox VE untuk memungkinkan pemindahan Virtual Machine antar node tanpa perlu menghentikan layanan yang sedang berjalan. Implementasi ini dilakukan dengan memanfaatkan konfigurasi cluster, jaringan, dan storage yang telah disiapkan sebelumnya.

Virtual Machine ditempatkan pada shared storage sehingga dapat diakses oleh seluruh node dalam cluster. Dengan kondisi tersebut, sistem mendukung proses live migration secara langsung antar node. Fitur ini menjadi salah satu bagian penting dalam memastikan fleksibilitas pengelolaan layanan pada lingkungan cluster.



Gambar 13 Proses Live Migration VM Antar-Node pada Cluster Proxmox VE

Live migration berhasil diimplementasikan pada sistem cluster Proxmox VE. Virtual Machine dapat dipindahkan dari satu node ke node lain dalam kondisi running, menandakan bahwa fitur live migration telah berfungsi dengan baik pada lingkungan cluster.



Gambar 14 Migrasi Sukses

Hasil:

- VM berpindah ke node tujuan
- Ping hanya delay singkat
- IP VM tetap sama

4.8 Implementasi High Availability (HA)

Fitur High Availability (HA) diimplementasikan pada sistem cluster Proxmox VE untuk memastikan layanan tetap berjalan ketika terjadi gangguan pada salah satu node. Implementasi HA dilakukan dengan mendaftarkan Virtual Machine ke dalam HA Group sehingga VM berada di bawah pengelolaan sistem HA.

Dengan implementasi ini, apabila node tempat VM berjalan mengalami kegagalan, sistem secara otomatis akan menjalankan kembali VM pada node lain yang tersedia di dalam cluster. Hal ini bertujuan untuk meningkatkan keandalan dan ketersediaan layanan pada lingkungan virtualisasi.

BAB V

PENGUJIAN DAN ANALISIS

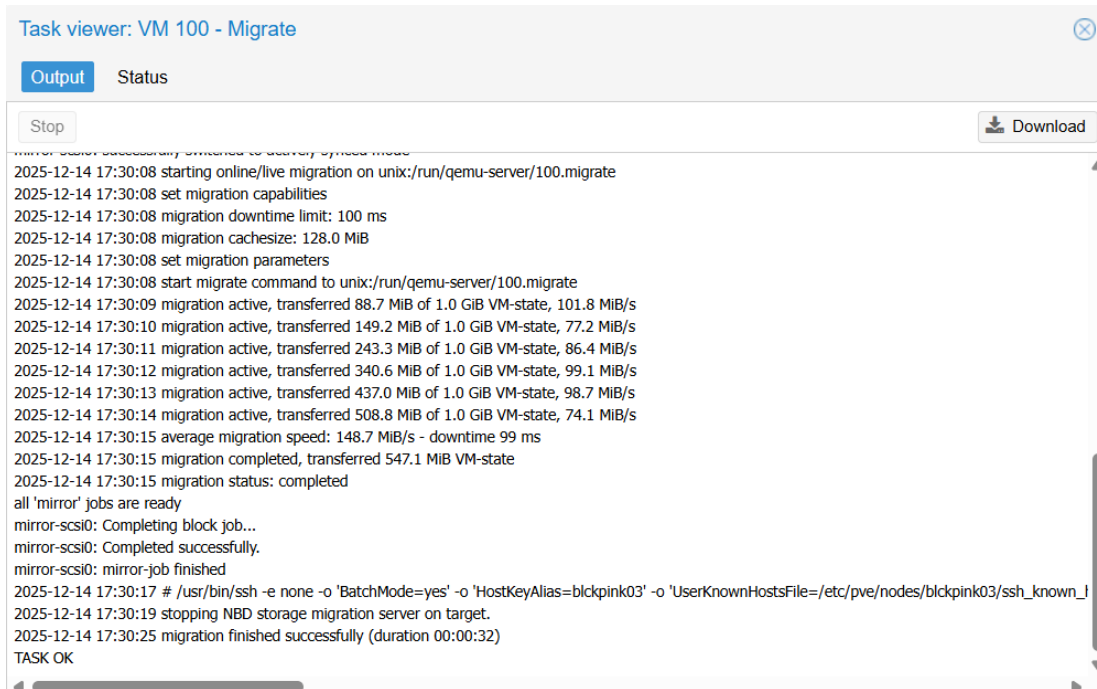
5.1 Analisis Konektivitas Jaringan antar Node dan VM

Langkah awal pengujian adalah memastikan seluruh entitas dalam cluster dapat saling berkomunikasi tanpa adanya hambatan (latency atau packet loss).

- Verifikasi Jaringan antar Node: Berdasarkan pengujian ICMP dari node blkcpink ke blkcpink02 (192.168.1.12), diperoleh hasil packet loss 0% dengan nilai RTT rata-rata \$0.489 ms\$.
- Verifikasi ke Node 3: Pengujian dari blkcpink02 ke blkcpink03 (192.168.1.13) juga menunjukkan koneksi yang stabil dengan latensi rata-rata \$0.667 ms\$.
- Aksesibilitas Virtual Machine: Node utama berhasil terhubung ke Virtual Machine 100 yang memiliki alamat IP 192.168.1.15.
- Analisis Jaringan: Nilai latensi yang sangat rendah (di bawah \$1 ms\$) menunjukkan bahwa infrastruktur jaringan yang dibangun memiliki bandwidth yang sangat mencukupi untuk menangani lalu lintas data Corosync (sinkronisasi cluster) dan proses migrasi data antar node tanpa gangguan.

5.2 Analisis Mendalam Live Migration (Manual Migration)

Pengujian dilakukan dengan memindahkan VM 100 dari satu node ke node lainnya saat sistem sedang berjalan (online migration).



Gambar 15 Analisis Migrasi Sukses

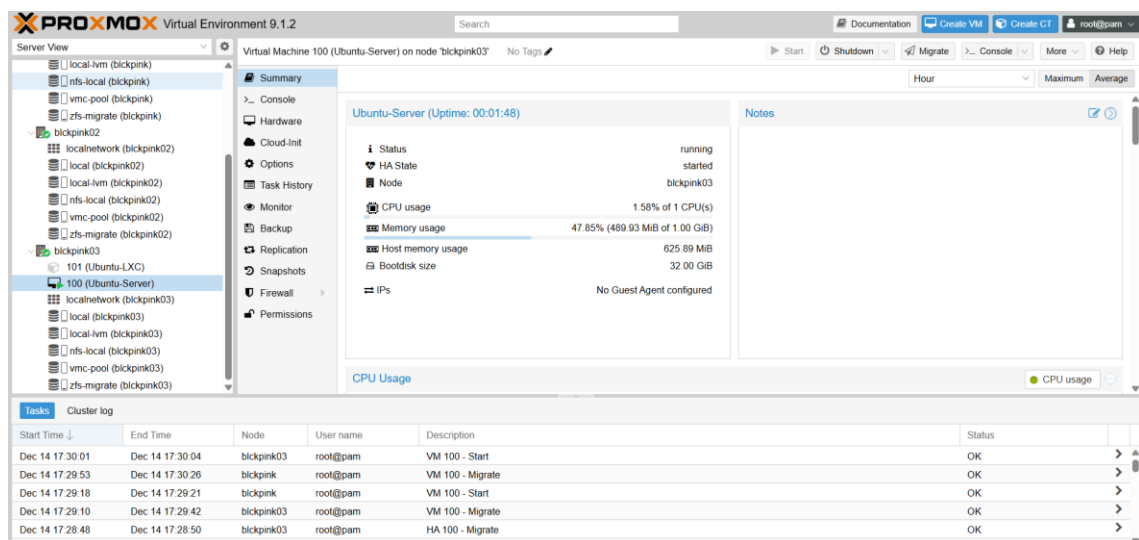
Analisis Log Migrasi:

- Downtime: Berdasarkan log sistem, migration downtime limit ditetapkan sebesar \$100 ms\$, namun sistem berhasil melakukan migrasi dengan actual downtime hanya \$99 ms\$.
- Transfer Data: Sistem memindahkan data VM-state sebesar \$547.1 MiB\$ dengan kecepatan rata-rata \$148.7 MiB/s\$.
- Durasi Total: Keseluruhan proses migrasi hingga status "TASK OK" memakan waktu selama \$32\$ detik\$.
- Kesimpulan Log: Keberhasilan migrasi dengan downtime di bawah \$0.1\$ detik\$ membuktikan bahwa Shared Storage NFS berfungsi dengan sangat baik, di mana sistem hanya perlu memindahkan isi RAM

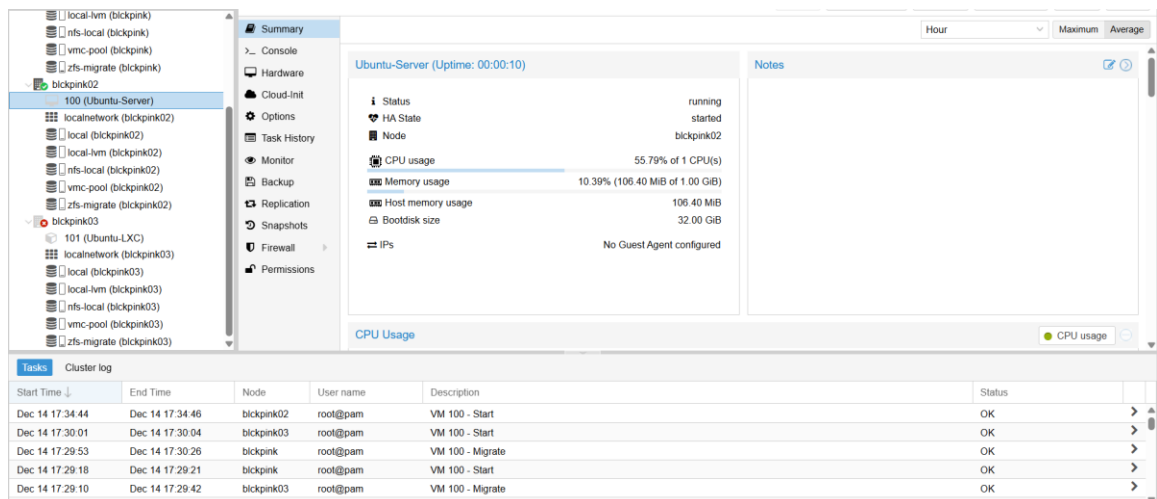
(memory state) tanpa harus menyalin seluruh disk image yang berukuran besar.

5.3 Analisis Mekanisme Failover dan High Availability (HA)

Pengujian ini bertujuan untuk membuktikan bahwa layanan tetap tersedia meskipun salah satu node mengalami kegagalan total (crash/power off).



Gambar 17 Before-Down-Node3



Gambar 16 After-Down-Node3

- Kondisi Awal: VM 100 berada dalam kondisi running dan started (HA aktif) di node blckpink03.
- Skenario Kegagalan: Node blckpink03 dimatikan secara paksa, yang ditandai dengan ikon "X" merah pada panel Server View.
- Hasil Failover Otomatis:
 1. Setelah node 3 terdeteksi offline, cluster manager secara otomatis memicu perintah "HA 100 - Migrate" dan "HA 100 - Start".
 2. VM 100 berhasil berpindah secara otomatis ke node blckpink02 dan kembali ke status running dalam waktu singkat.
 3. Analisis Analitik: Berdasarkan log Tasks, terlihat bahwa proses "HA 100 - Start" berhasil dieksekusi dengan status OK. Hal ini menunjukkan bahwa kebijakan HA yang telah diatur pada Proxmox berhasil mendeteksi kegagalan node dan melakukan pemulihan layanan (recovery) secara otomatis tanpa intervensi manual dari administrator.

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada Proyek Membangun Private Cloud Cluster dengan Proxmox VE ini, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Keberhasilan Pembentukan Cluster: Infrastruktur Private Cloud berhasil dibangun menggunakan tiga node Proxmox (blckpink01, blckpink02, dan blckpink03) yang terintegrasi secara sempurna dalam satu manajemen terpusat (Datacenter).
2. Peran Krusial Shared Storage: Penggunaan NFS sebagai shared storage terbukti menjadi komponen kunci yang memungkinkan fitur Live Migration berjalan efektif. Dengan shared storage, pemindahan VM antar node hanya memerlukan sinkronisasi state memori tanpa harus menyalin data disk yang besar.
3. Efektivitas High Availability (HA): Pengujian failover membuktikan bahwa fitur HA berfungsi dengan baik. Saat salah satu node (blckpink03) mengalami kegagalan/mati, sistem secara otomatis mendeteksi status offline dan membangkitkan kembali VM di node lain yang aktif (blckpink02) dalam waktu singkat.
4. Minimal Downtime: Fitur Live Migration manual berhasil dilaksanakan dengan actual downtime yang sangat rendah, yakni hanya sebesar 99 ms, sehingga layanan di dalam VM tetap dapat diakses oleh pengguna hampir tanpa interupsi.

6.2 Saran

Untuk pengembangan infrastruktur Private Cloud yang lebih optimal di masa depan, beberapa saran yang dapat dipertimbangkan adalah:

1. Redundansi Jaringan: Disarankan untuk menggunakan teknik Network Bonding (menggabungkan dua atau lebih kartu jaringan) pada jalur Corosync dan Migration untuk menghindari kegagalan cluster jika salah satu kabel jaringan terputus.
2. Optimasi Shared Storage: Mengingat NFS sangat bergantung pada satu server storage, penggunaan Ceph Storage (distribusi storage berbasis cluster) bisa menjadi alternatif yang lebih baik untuk menghilangkan Single Point of Failure pada sisi penyimpanan.
3. Implementasi Fencing: Untuk mencegah kondisi Split-Brain pada cluster HA, disarankan menambahkan perangkat Fencing (seperti hardware watchdog atau PDU pintar) agar manajemen node lebih aman saat terjadi ketidaksinkronan status antar node.

DAFTAR PUSTAKA

- Debian Project. (2025). *Debian GNU/Linux installation guide*.
<https://www.debian.org/releases/stable/installmanual>
- Linux Kernel Organization. (2025). *The Linux kernel archives: Version 6.17*.
<https://www.kernel.org/>
- Proxmox Server Solutions GmbH. (2025). *Proxmox virtual environment administration guide*. <https://pve.proxmox.com/pve-docs/pve-admin-guide.html>
- Proxmox Wiki. (2025). *Cluster manager*.
https://pve.proxmox.com/wiki/Cluster_Manager
- Proxmox Wiki. (2025). *High availability*.
https://pve.proxmox.com/wiki/High_Availability
- Proxmox Wiki. (2025). *QEMU/KVM virtual machines*.
https://pve.proxmox.com/wiki/Qemu/KVM_Virtual_Machines
- Proxmox Wiki. (2025). *Storage: NFS*. https://pve.proxmox.com/wiki/Storage:_NFS

LAMPIRAN

A. Konfigurasi (System Configurations)

A.1 /etc/exports – Konfigurasi NFS Server

```
GNU nano 8.4 /etc/exports *
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4 gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes gss/krb5i(rw,sync,no_subtree_check)
#
/srv/nfs/proxmox 192.168.1.0/24(rw,sync,no_subtree_check,no_root_squash)

Write to File: /etc/exports
^G Help      M-I DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

A.2 /etc/exports – Konfigurasi NFS Server (Backup)

```
GNU nano 6.2 /etc/exports *
/srv/nfs4/backups 192.168.137.0/24(rw,sync,no_subtree_check)
[]

^G Help      ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit      ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

B. Kode

B.1 Perintah instalasi dan konfigurasi NFS Server

```
apt update  
apt install nfs-kernel-server -y  
mkdir -p /mnt/nfs-share  
chown -R nobody:nogroup /mnt/nfs-share  
systemctl enable nfs-server  
systemctl restart nfs-server
```

B.2 Perintah pembuatan dataset ZFS untuk storage pool

```
zfs create blckpink-storage-pool/vmc  
zfs create blckpink-storage-pool/iso  
zfs create blckpink-storage-pool/backup
```

B.3 Perintah pembuatan ZFS pool dari disk /dev/sdb

```
zpool create blckpink-storage-pool /dev/sdb
```

C. SCREENSHOT

C.1 Storage NFS

Add: NFS

General Backup Retention

ID: Nodes: ✕ ▼

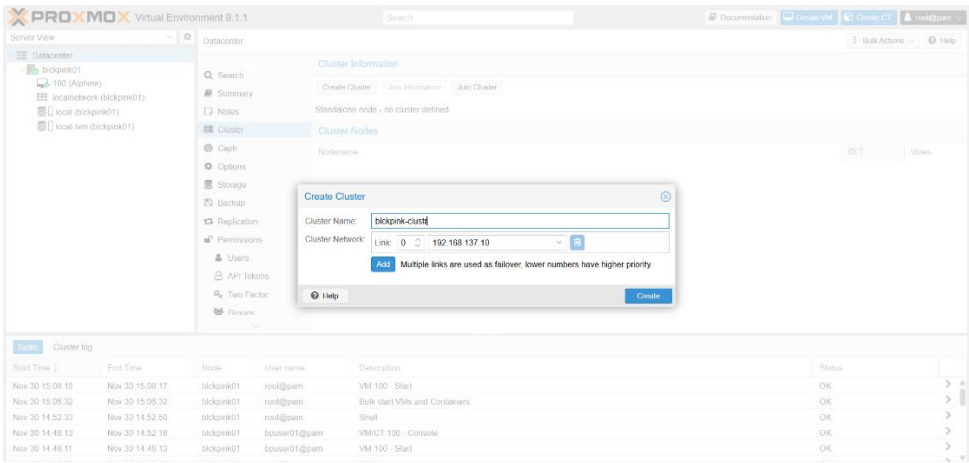
Server: Enable: ☒

Export: ▼

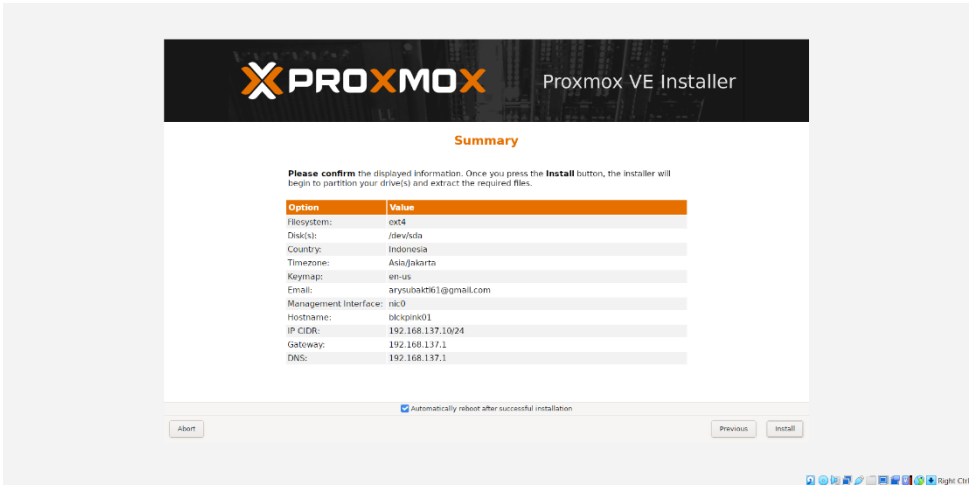
Content: ▼

? Help Advanced ☐ Add

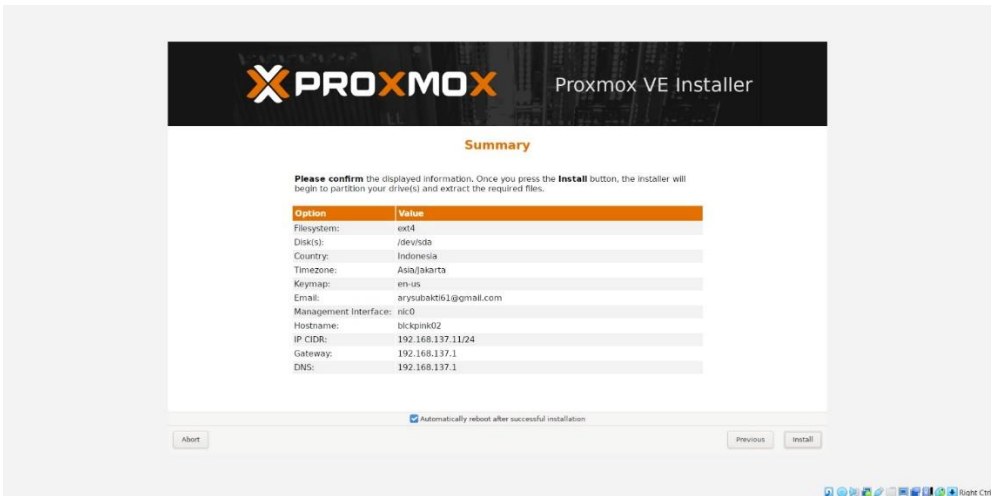
C.2 Cluster-Create-Node1



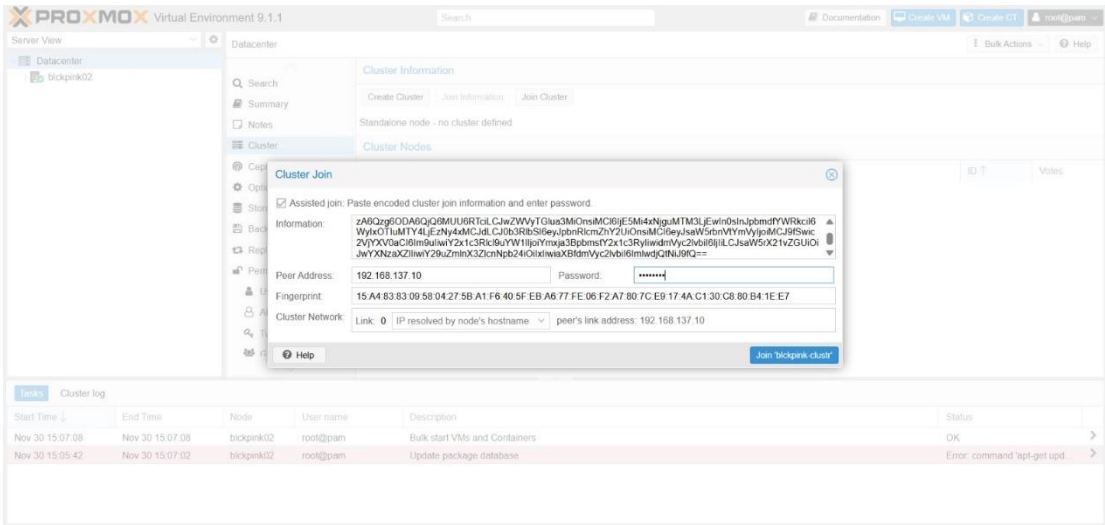
C.3 Config-node1



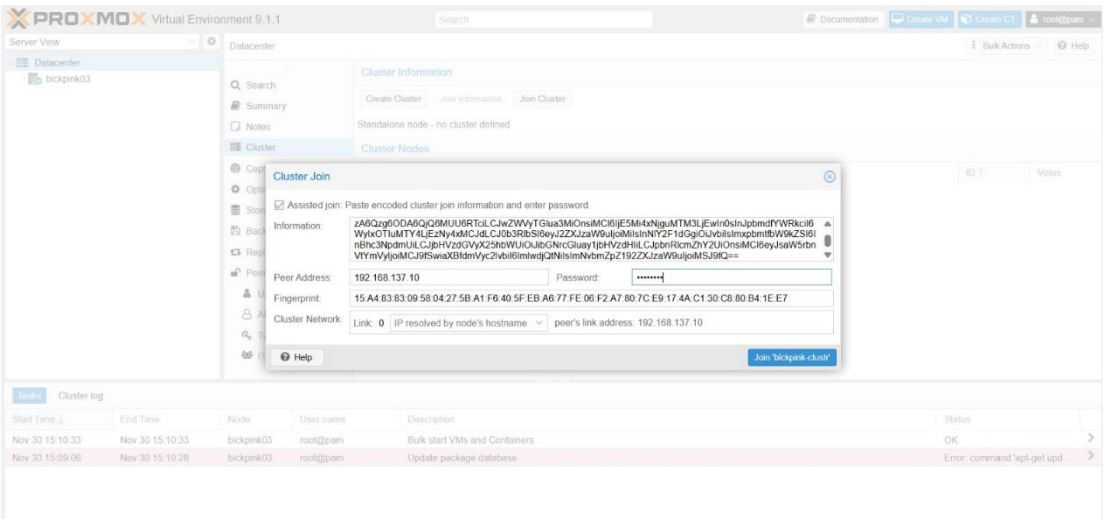
C.4 Config-Node2



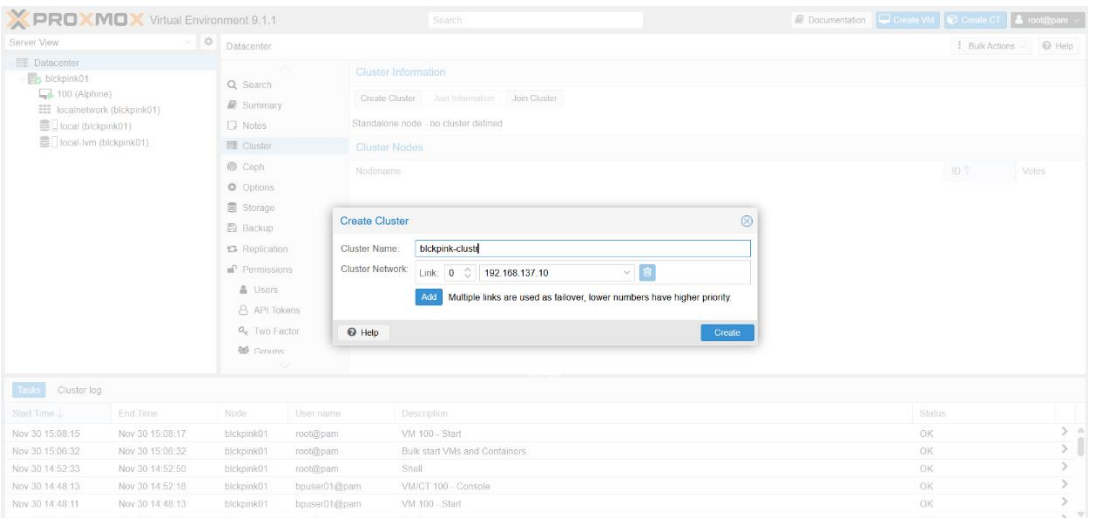
C.5 Join-Cluster-Node2



C.6 Join-Cluster-Node3



C.7 Cluster



C.8 Output zfs list – Daftar dataset ZFS yang telah dibuat

```
root@blckpink:~# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
blckpink-storage-pool              680K  364G   104K   /blckpink-storage-pool
blckpink-storage-pool/backup        96K   364G    96K   /blckpink-storage-pool/backup
blckpink-storage-pool/iso           96K   364G    96K   /blckpink-storage-pool/iso
blckpink-storage-pool/vmc           96K   364G    96K   /blckpink-storage-pool/vmc
root@blckpink:~#
```

C.9 Proses pembuatan dataset ZFS dan verifikasi

```
root@blckpink:~# zfs create blckpink-storage-pool/vmc
root@blckpink:~# zfs list
NAME                                USED  AVAIL  REFER  MOUNTPOINT
blckpink-storage-pool              468K  364G    96K   /blckpink-storage-pool
blckpink-storage-pool/vmc          96K   364G    96K   /blckpink-storage-pool/vmc
root@blckpink:~# zfs create blckpink-storage-pool/backup
root@blckpink:~# zfs create blckpink-storage-pool/iso
root@blckpink:~#
```


C.10 Antarmuka Proxmox – Menambahkan storage directory

Add: Directory

General

Backup Retention

ID:	<input type="text" value="vmc-pool"/>	Nodes:	<input type="text" value="All (No restrictions)"/>
Directory:	<input type="text" value="/blckpink-storage-pool/vmc"/>	Enable:	<input checked="" type="checkbox"/>
Content:	<input type="text" value="Disk image, Container"/>	Shared:	<input type="checkbox"/>

Help

Advanced

Add

C.11 Uji koneksi jaringan (ping) ke 192.168.1.12

```
Linux blckpink 6.17.2-2-pve #1 SMP PREEMPT_DYNAMIC PMX 6.17.2-2 (2025-11-26T12:33Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@blckpink:~# ping 192.168.1.12
PING 192.168.1.12 (192.168.1.12) 56(84) bytes of data.
64 bytes from 192.168.1.12: icmp_seq=1 ttl=64 time=0.436 ms
64 bytes from 192.168.1.12: icmp_seq=2 ttl=64 time=0.498 ms
64 bytes from 192.168.1.12: icmp_seq=3 ttl=64 time=0.533 ms
^C
--- 192.168.1.12 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.436/0.489/0.533/0.040 ms
root@blckpink:~#
```

C.12 Uji koneksi jaringan (ping) ke 192.168.1.15

```
Linux blkcpink 6.17.2-2-pve #1 SMP PREEMPT_DYNAMIC PMX 6.17.2-2 (2025-11-26T12:33Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@blkcpink:~# ping 192.168.1.15
PING 192.168.1.15 (192.168.1.15) 56(84) bytes of data.
64 bytes from 192.168.1.15: icmp_seq=1 ttl=64 time=1.30 ms
64 bytes from 192.168.1.15: icmp_seq=2 ttl=64 time=0.805 ms
64 bytes from 192.168.1.15: icmp_seq=3 ttl=64 time=0.863 ms
64 bytes from 192.168.1.15: icmp_seq=4 ttl=64 time=0.813 ms
^C
--- 192.168.1.15 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 0.805/0.944/1.296/0.204 ms
root@blkcpink:~#
```

C.13 Uji koneksi dari node blkcpink02 ke 192.168.1.13

```
Linux blkcpink02 6.17.2-1-pve #1 SMP PREEMPT_DYNAMIC PMX 6.17.2-1 (2025-10-21T11:55Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sun Dec 14 17:25:41 2025 from 192.168.1.10
root@blkcpink02:~# ping 192.168.1.13
PING 192.168.1.13 (192.168.1.13) 56(84) bytes of data.
64 bytes from 192.168.1.13: icmp_seq=1 ttl=64 time=0.381 ms
64 bytes from 192.168.1.13: icmp_seq=2 ttl=64 time=0.390 ms
64 bytes from 192.168.1.13: icmp_seq=3 ttl=64 time=1.23 ms
^C
--- 192.168.1.13 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2075ms
rtt min/avg/max/mdev = 0.381/0.667/1.231/0.398 ms
root@blkcpink02:~#
```