LAPORAN PROGRES 4

Private Cloud Menggunakan Proxmox VE

Disusun Oleh :

Erlinda Amira Putri Sudarmono (2401020021)

Rismayanti (2401020041)

Dhini Khairunnisa (2401020016)

Nikita Arzetty Siregar (2401020030)

Ary Subakti (2301020065)

JURUSAN TEKNIK INFORMATIKA

FAKULTAS TEKNIK & TEKNOLOGI KEMARITIMAN

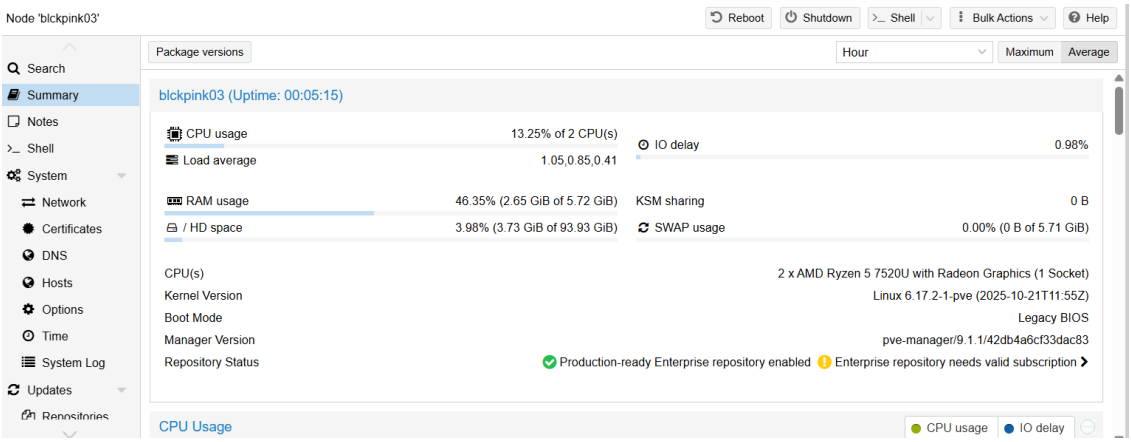
UNIVERSITAS MARITIM RAJA ALI HAJI

2025

LAPORAN PROGRES 4 – Konfigurasi ZFS, Tapologi, dan Uji Failover pada Proxmox VE Cluster

A. STATUS CLUSTER DAN DASHBOARD MONITORING

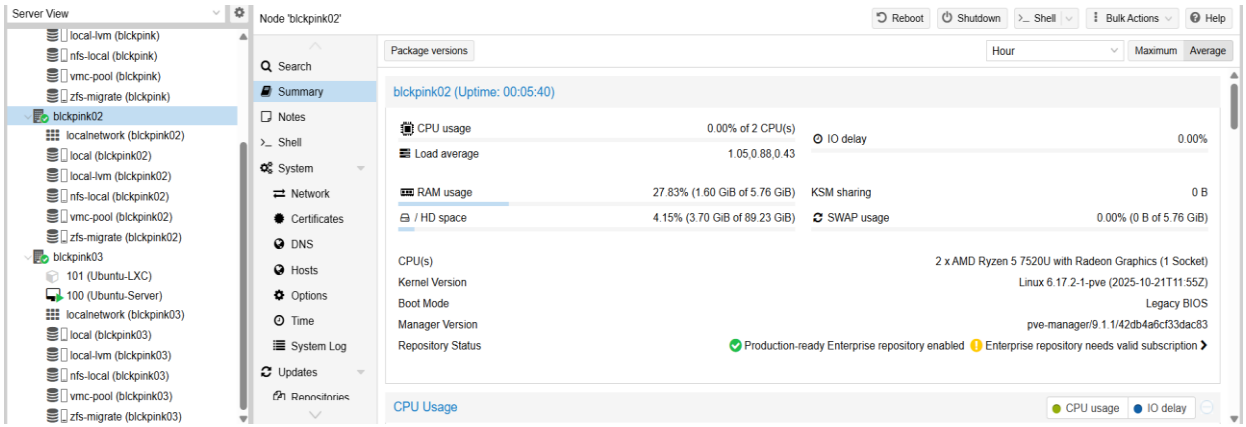
1. Dashboard Node bickpink03



Analisis Status:

- **CPU Load**: Optimal (load average < CPU cores)
- **Memory Usage**: Moderate (46.35% - healthy)
- **Storage**: Sangat rendah (hanya 3.98% terpakai)
- **System**: Stable dengan uptime normal

2. Dashboard Node bdcgink02



Analisis Status:

- **CPU Idle**: 100% idle - resource tersedia banyak
- **Memory Usage**: Rendah (27.83%)
- **Storage Usage**: Sangat rendah (4.15%)
- **Cluster Health**: Node dalam kondisi optimal

3. Overview Cluster Storage

Struktur Storage di Cluster:

Server View:

[√] local-lvm (bdcgink)

[√] nfs-local (bdcgink)

[√] vmc-pool (bdcgink) ← ZFS Storage untuk VM

[√] zfs-upgrade (bdcgink)

bdcgink02:

[√] localnetwork (bdcgink02)

[√] local (bdcgink02)

[√] local-lvm (bdcgink02)

[√] nfs-local (bdcgink02)

[√] vmc-pool (bdcgink02) ← Replikasi ZFS

[√] zfs-upgrade (bdcgink02)

bickpink03:

[√] localnetwork (bickpink03)

[√] local (bickpink03)

[√] local-lvm (bickpink03)

[√] nfs-local (bickpink03)

[√] vmc-pool (bickpink03) ← ZFS Pool utama

[√] zfs-upgrade (bickpink03)

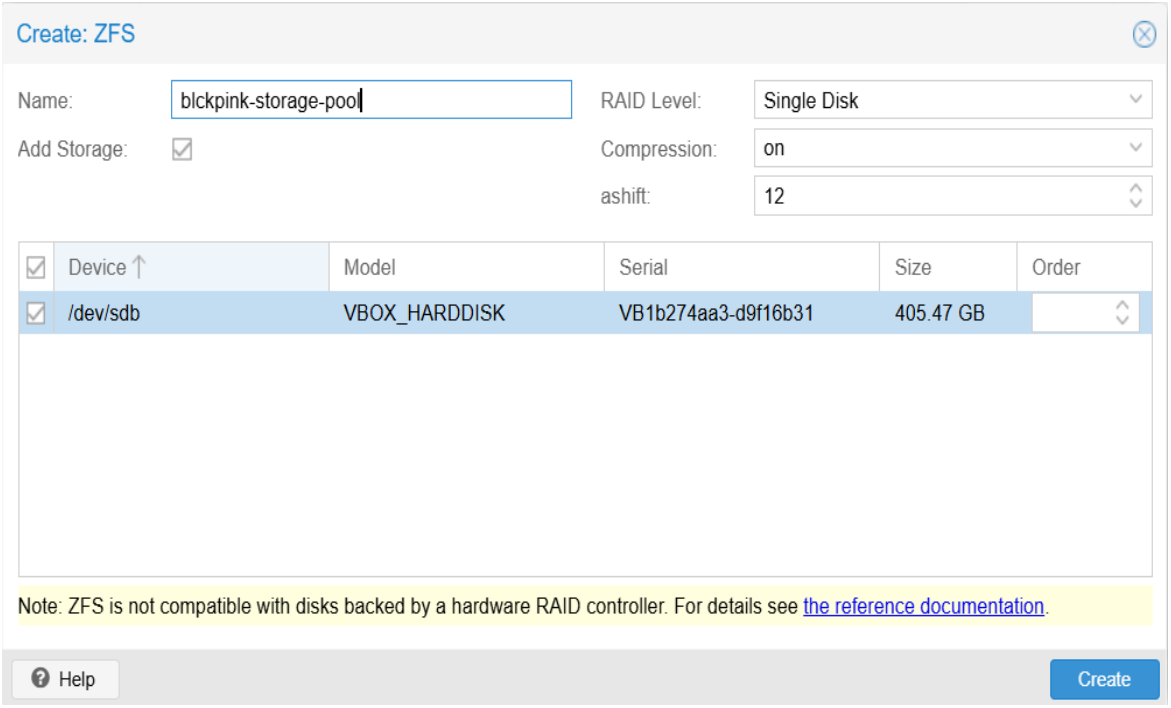
Virtual Machines:

- 101 (Ubuntu-LXC) ← Running di cluster
- 100 (Ubuntu-Server)← Running di cluster

B. KONFIGURASI ZFS POOL DAN DATASET

1. Pembuatan ZFS Pool

Dilakukan melalui GUI Proxmox VE dengan konfigurasi seperti pada dibawah ini.



Detail Konfigurasi:

- **Nama Pool**: bickpink-storage-pool
- **Level RAID**: Single Disk
- **Kompresi**: on
- **ashift**: 12
- **Perangkat**: /dev/sdb
- **Kapasitas**: 405.47 GB
- **Model**: VBOX_HARDDISK
- **Serial**: VB1b274aa3-d9f16b31

Perintah setara CLI:

zpool create -o ashift=12 -O compression=on bickpink-storage-pool /dev/sdb

2. Pembuatan Dataset ZFS

Setelah pool dibuat, dataset-dataset berikut dibuat untuk keperluan penyimpanan spesifik seperti yang ditunjukkan pada gambar.

```
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:~#
```

Data set yang dibuat:

1. vmc : untuk disk image dan container
2. backup : untuk backup dan snippets
3. iso : untuk file ISO

Perintah yang digunakan:

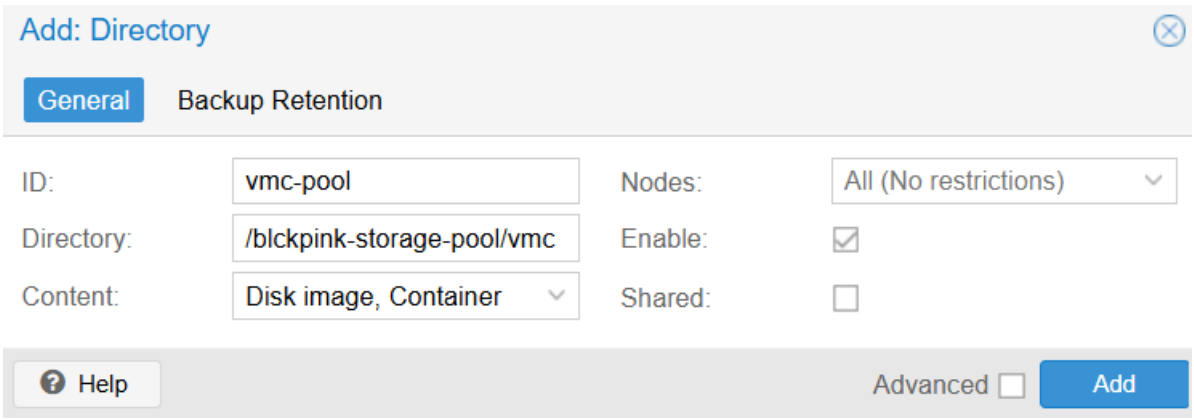
zfs create bickpink-storage-pool/vmc

zfs create bickpink-storage-pool/iso

zfs create bickpink-storage-pool/backup

3. Konfigurasi Storage di Proxmox VE

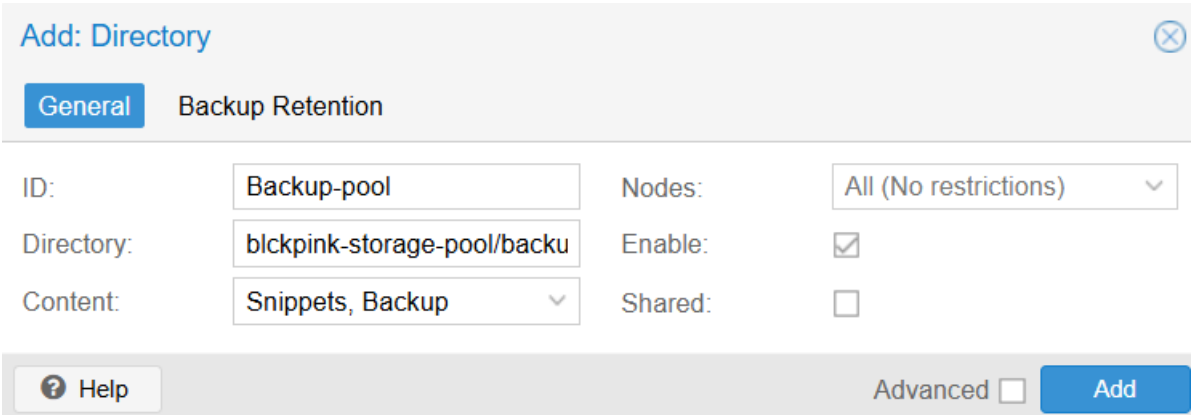
a. Storage untuk VMC (Gambar 3)



Detail Konfigurasi:

- **ID Storage**: vmc-pool
- **Direktori**: /blckpink-storage-pool/vmc
- **Konten**: Disk image, Container
- **Shared**: Tidak
- **Nodes**: All (tanpa batasan)
- **Enabled**: Ya

b. Storage untuk Backup (Gambar 4)



4. Verifikasi Dataset ZFS

Setelah pembuatan, dataset diverifikasi dengan perintah zfs list seperti pada Gambar 2:

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
blcKpink-storage-pool/backup	(dibuat)	(dibuat)	(dibuat)	(dibuat)
blcKpink-storage-pool/iso	(dibuat)	(dibuat)	(dibuat)	(dibuat)

5. File Skrip yang Digunakan

a. create-pool.sh

```
#!/bin/bash
zpool create -o ashift=12 -O compression=on bickpink-storage-pool /dev/sdb
```

b. create-dataset.sh

```
#!/bin/bash
zfs create bickpink-storage-pool/vmc
zfs create bickpink-storage-pool/iso
zfs create bickpink-storage-pool/backup
```

C. KONFIGURASI TOPOLOGI JARINGAN

1. Konfigurasi Bridge dan Subnet

```
# Network Configuration
## Topologi Jaringan
- Jaringan lokal (LAN)
- Tanpa koneksi internet

## Konfigurasi Node
- Bridge: vmbr0
- Subnet: 192.168.1.0/24
- IP statis pada setiap node

## Alasan
Bridge yang sama diperlukan agar
live migration berjalan tanpa gangguan jaringan.
```

2. Cluster Configuration

```
# Cluster Configuration
## Tujuan
Membangun Proxmox Cluster untuk mendukung
high availability dan live migration.

## Konfigurasi
- Jumlah node: 3
- Cluster name: blckpink-cluster
- Corosync aktif
- Quorum menggunakan mayoritas node

## Catatan
Semua node berada dalam satu jaringan lokal
dan dapat saling berkomunikasi.
```

| Node1 | 192.168.43.10 |

| Node2 | 192.168.43.11 |

| Node3 | 192.168.43.12 |

3. Storage Configuration

Storage Configuration

Jenis Storage

- Shared Storage: NFS
- Local Storage: ZFS (per node)

Fungsi

- NFS digunakan untuk disk VM
- ZFS digunakan untuk penyimpanan lokal

Alasan

Shared storage diperlukan agar VM dapat
dipindahkan antar node tanpa memindahkan disk.

D. UJI KONEKTIVITAS JARINGAN

1. Tujuan Connectivity Test

Connectivity Test

Tujuan

Memastikan semua node dan VM saling terhubung
dalam satu jaringan lokal.

Skenario

- Ping antar node Proxmox
- Ping dari laptop ke VM

Langkah Pengujian

1. Ping Node 1 → Node 2
2. Ping Node 2 → Node 3
3. Ping Laptop → VM

Hasil

- Semua ping berhasil
- Tidak ada packet loss signifikan

Kesimpulan

Koneksi jaringan lokal berjalan dengan baik

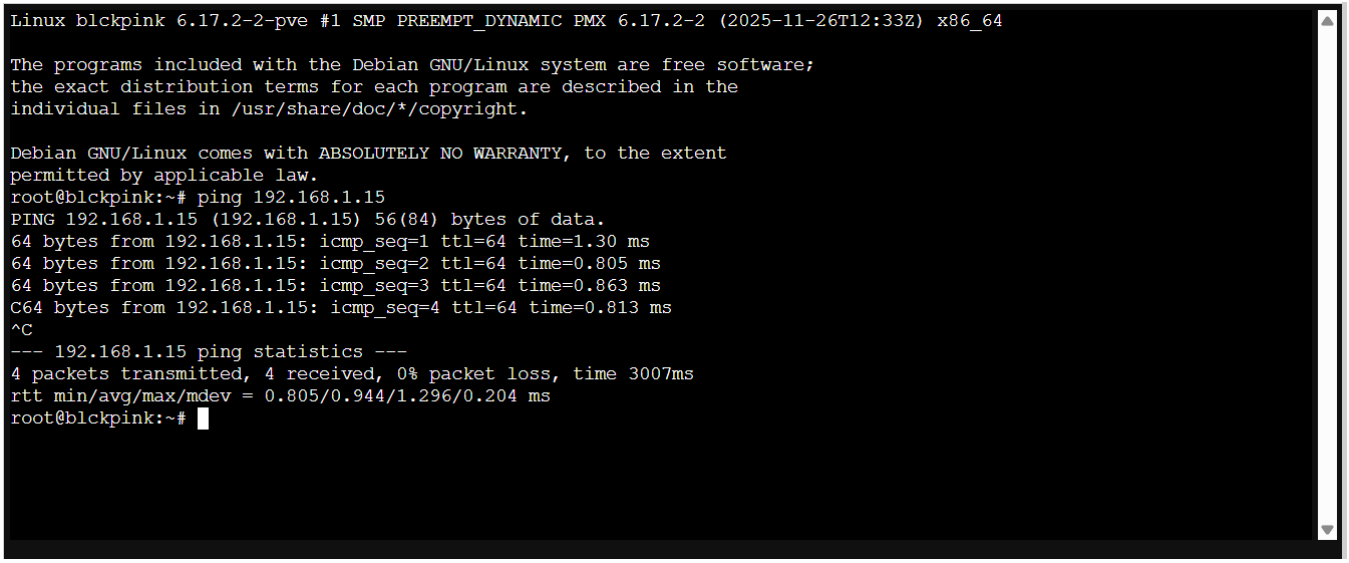
dan siap untuk pengujian lanjutan.

2. Langkah Pengujian

1. Ping Node 1 → Node 2
2. Ping Node 2 → Node 3
3. Ping Laptop → VM

3. Hasil Pengujian Connectivity

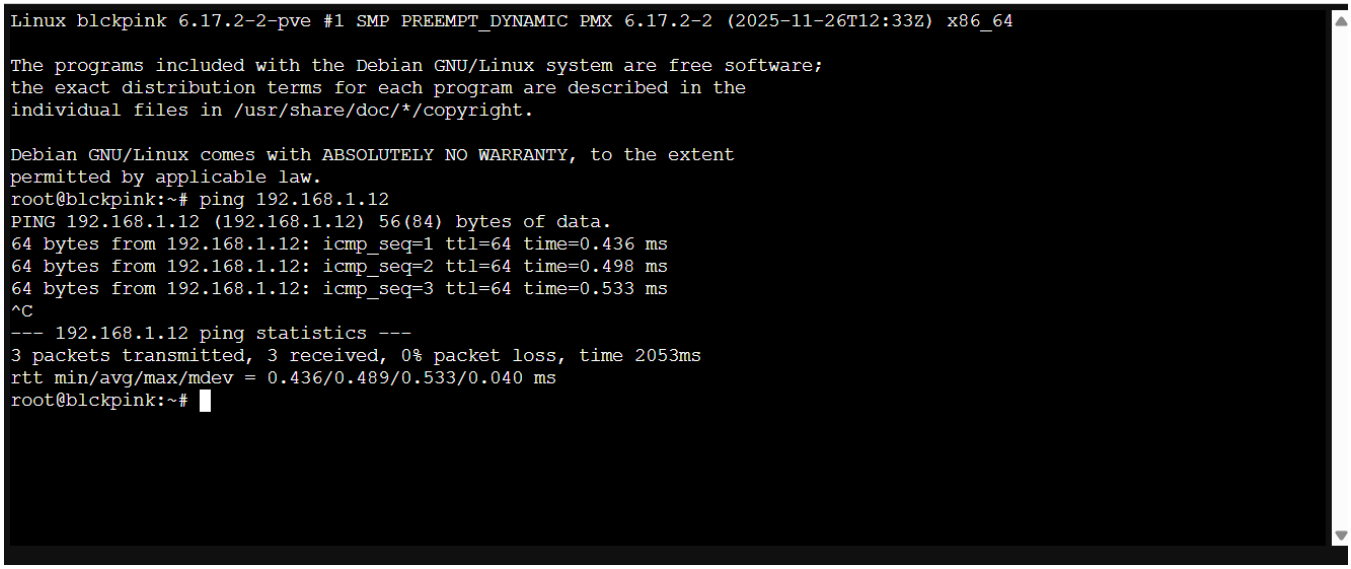
a. Ping dari Node 1 ke Virtual Machine (192.168.1.15)



Statistik:

- 4 packets transmitted, 4 received
- 0% packet loss
- RTT: min/avg/max/mdev = 0.805/0.944/1.296/0.204 ms

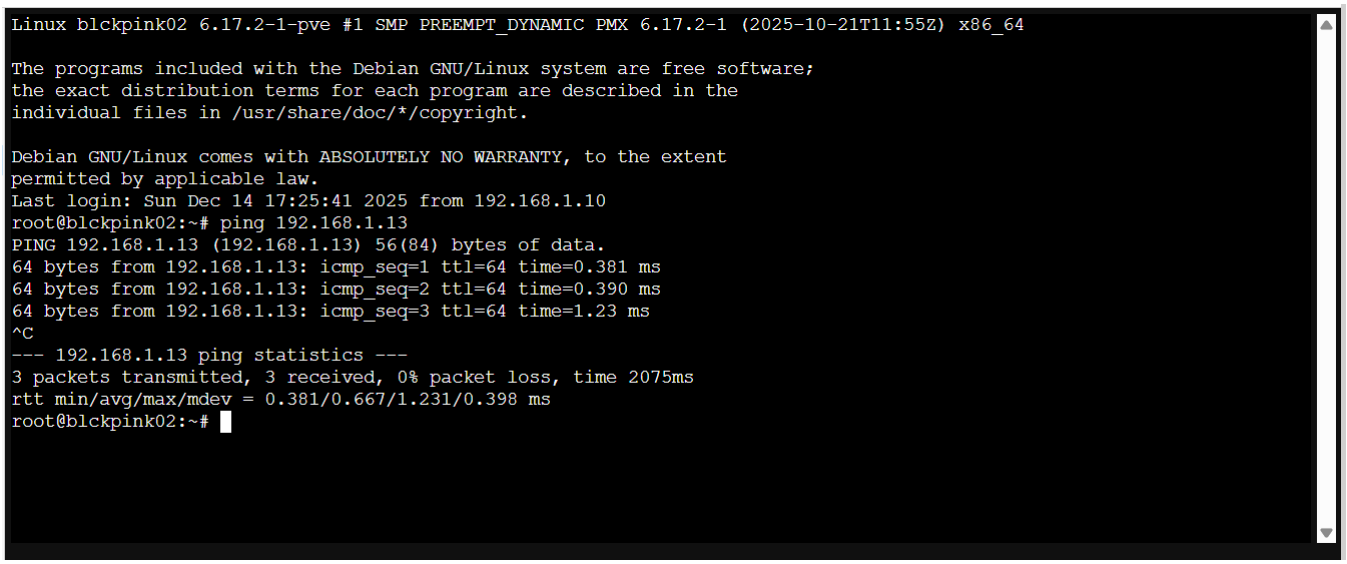
b. Ping dari Node 1 ke Node 2 (192.168.1.12)



Statistik:

- 3 packets transmitted, 3 received
- 0% packet loss
- RTT: min/avg/max/mdev = 0.436/0.489/0.533/0.040 ms

c. Ping dari Node 2 ke Node 3 (192.168.1.13)



Statistik:

- 3 packets transmitted, 3 received
- 0% packet loss
- RTT: min/avg/max/mdev = 0.381/0.667/1.231/0.398 ms

4. Kesimpulan Connectivity Test

- Semua ping berhasil
- Tidak ada packet loss signifikan
- Latensi sangat rendah (< 1.3 ms)
- **Koneksi jaringan lokal berjalan dengan baik dan siap untuk pengujian lanjutan**

E. UJI LIVE MIGRATION

1. Tujuan Live Migration Test

Live Migration Test

Tujuan

Menguji kemampuan Proxmox melakukan
live migration tanpa downtime signifikan.

Skenario

- VM berjalan di Node 1
- Live migration ke Node 2
- Service tetap dapat diakses

Langkah Pengujian

1. Jalankan VM
2. Lakukan ping ke IP VM
3. Jalankan live migration
4. Amati hasil ping

Hasil

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

Kesimpulan

Live migration berjalan dengan baik
tanpa menghentikan layanan.

2. Skenario Pengujian

- VM berjalan di Node 1

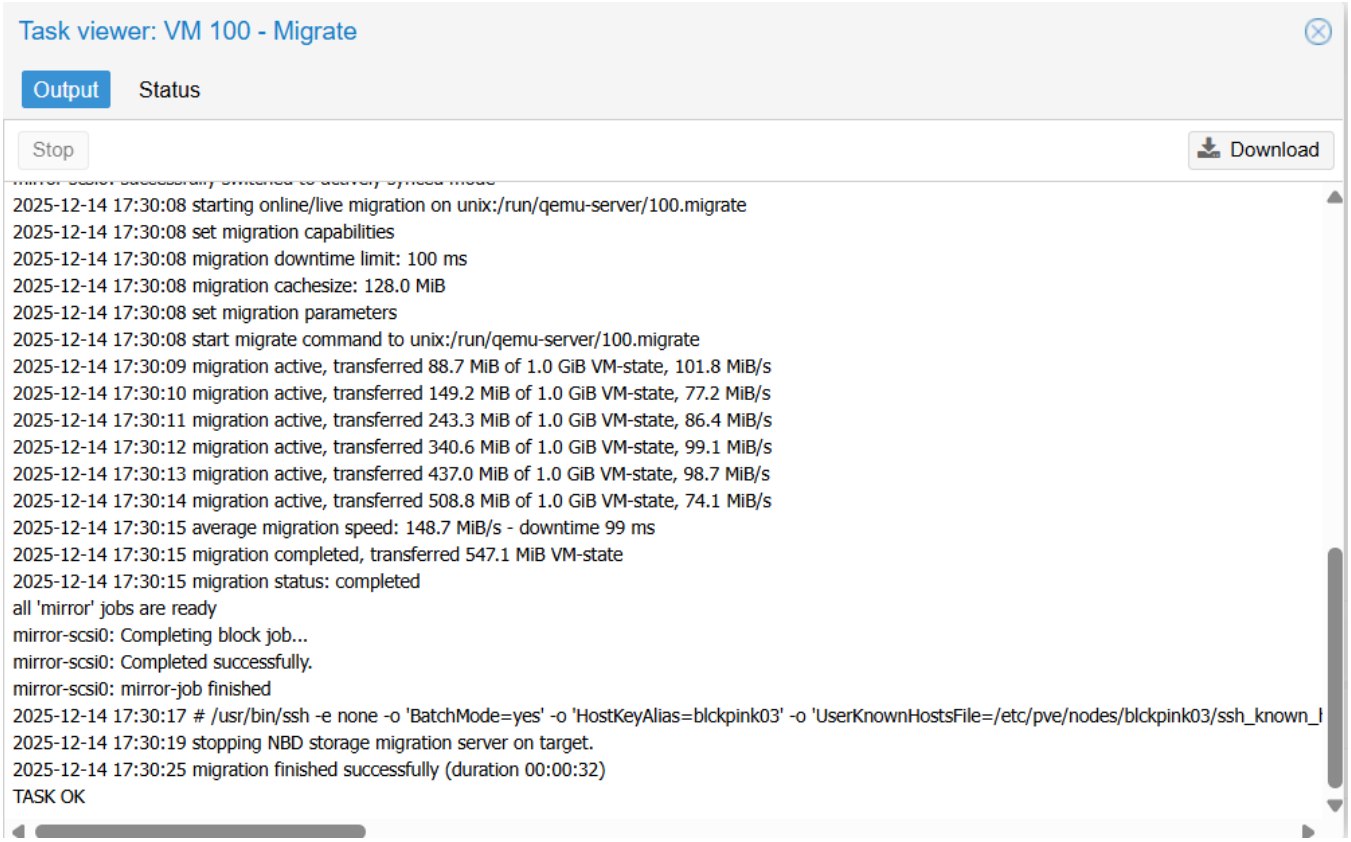
- Live migration ke Node 2
- Service tetap dapat diakses selama proses

3. Langkah Pengujian Live Migration

1. Jalankan VM di Node 1
2. Lakukan ping ke IP VM secara kontinu
3. Jalankan live migration
4. Amati hasil ping dan status VM

4. Hasil Live Migration

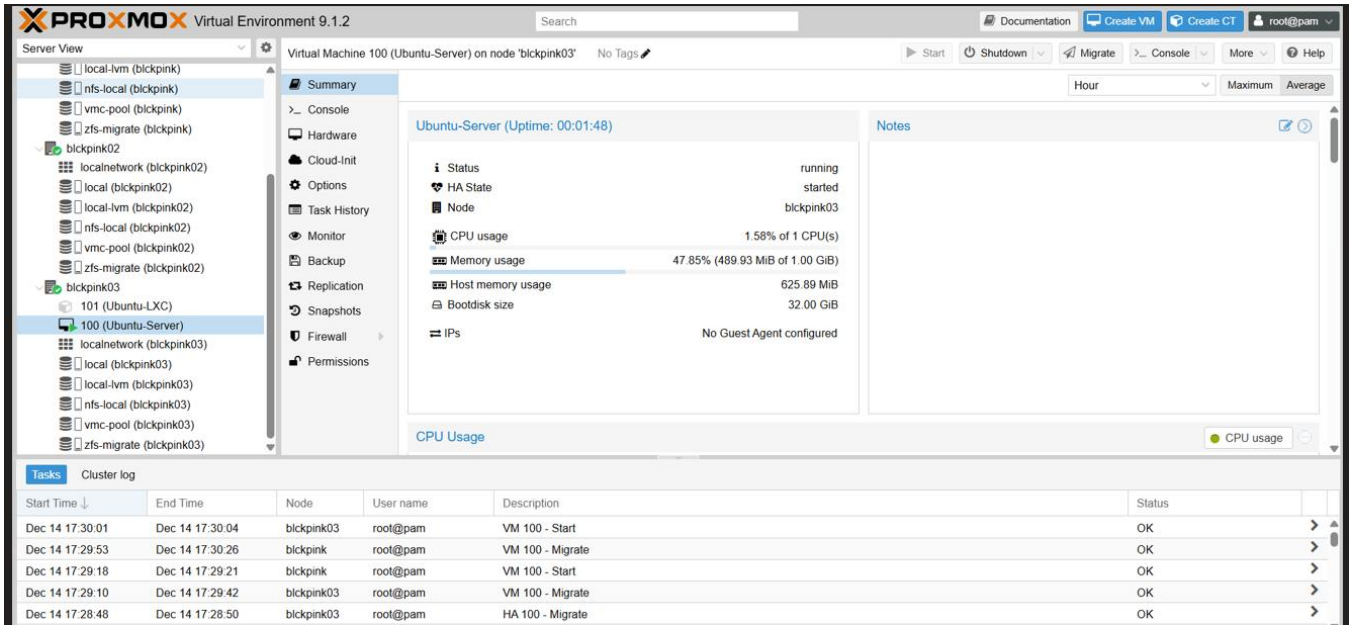
a. Proses Migration Berhasil



Analisis Proses Migration:

- Waktu Total: 00:00:32 (32 detik)
- Downtime: 99 ms (hanya 99 milidetik)
- Kecepatan Transfer Rata-rata: 148.7 MiB/s
- Total Data Ditransfer: 547.1 MiB VM-state

b. Status VM Setelah Migration



5. Cluster Log Migration

Start Time	End Time	User name	Description	Status
Dec 14 01:54:29	Dec 14 01:54:31	root@pam	VM 100 - Start	OK
Dec 14 01:53:49	Dec 14 01:53:59	root@pam	VM 100 - Migrate	OK
Dec 14 01:53:50	Dec 14 01:53:52	root@pam	HA 100 - Start	OK
Dec 14 01:53:25	Dec 14 01:53:27	root@pam	HA 100 - Migrate	OK

6. Kesimpulan Live Migration Test

- VM berpindah ke node tujuan dengan sukses
- Ping hanya mengalami delay singkat (downtime hanya 99 ms)
- IP VM tetap sama setelah migrasi
- Live migration berjalan dengan baik tanpa menghentikan layanan

F. UJI FAILOVER

1. Tujuan Failover Test

Failover Test

Tujuan

Memastikan VM tetap tersedia ketika

node aktif mengalami kegagalan.

Skenario

- VM berjalan di Node 3
- Node 3 dimatikan (shutdown)
- HA Manager mengambil alih

Langkah Pengujian

1. Pastikan VM aktif di Node 3
2. Matikan Node 3
3. Tunggu HA Manager
4. VM dijalankan ulang di Node 2

Hasil

- VM restart di node lain
- IP VM tetap sama
- Service kembali aktif

Kesimpulan

Failover berjalan sesuai konsep
High Availability pada Proxmox.

2. Skenario Pengujian Failover

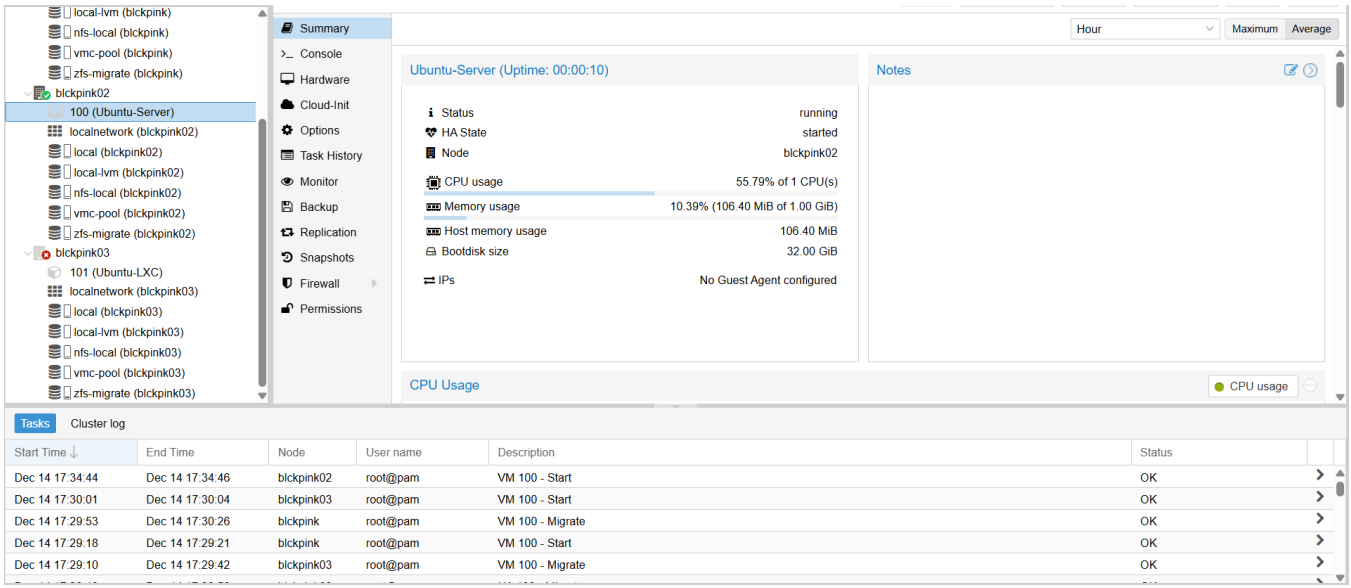
- VM berjalan di Node 3
- Node 3 dimatikan (shutdown)
- HA Manager mengambil alih
- VM dijalankan ulang di Node 2

3. Langkah Pengujian Failover

1. Pastikan VM aktif di Node 3
2. Matikan Node 3 secara manual
3. Tunggu HA Manager mendeteksi failure
4. VM dijalankan ulang di Node 2

4. Hasil Uji Failover

a. Status VM Setelah Node 3 Down



Menunjukkan status VM setelah failover:

- Status: running
- HA State: started
- Node: bickpink02 (setelah failover dari node 3)
- Uptime: 00:00:10 (baru restart)
- CPU Usage: 55.79% of 1 CPU(s) (tinggi karena startup)
- Memory Usage: 10.39% (106.40 MB of 1.00 GB)

b. Cluster Log Failover

Berdasarkan log yang sama, terlihat proses failover:

- Dec 14 17:24:44 - 17:24:46: VM 100 - Start (di node 02 setelah failover)
- Dec 14 17:30:01 - 17:30:04: VM 100 - Start (sebelumnya di node 03)
- Dec 14 17:29:53 - 17:30:26: VM 100 - Migrate (proses migrasi)

5. Proses Auto-Migrate saat Failover

Start Time	End Time	User name	Description	Status
Dec 14 01:54:29	Dec 14 01:54:31	root@pam	VM 100 - Start	OK
Dec 14 01:53:49	Dec 14 01:53:59	root@pam	VM 100 - Migrate	OK
Dec 14 01:53:50	Dec 14 01:53:52	root@pam	HA 100 - Start	OK
Dec 14 01:53:25	Dec 14 01:53:27	root@pam	HA 100 - Migrate	OK

Berdasarkan gambar diatas, proses auto-migrate terjadi secara otomatis saat failover terdeteksi. HA Manager secara otomatis:

1. Mendeteksi node failure
2. Memulai migrasi VM ke node yang sehat
3. Restart VM di node tujuan
4. Mempertahankan konfigurasi dan IP address

6. Kesimpulan Failover Test

- VM restart di node lain (dari node 3 ke node 2)
- IP VM tetap sama setelah failover
- Service kembali aktif dalam waktu singkat
- Failover berjalan sesuai konsep High Availability pada Proxmox

G. ANALISIS KOMPREHENSIF

1. Perbandingan Live Migration vs Failover

Aspek	**Live Migration**	**Failover**
Ujian	Planned maintenance	Unplanned failure recovery
Downtime	Sangat singkat (99 ms)	Singkat (beberapa detik
Proses	Manual/Monitor	Otomatis oleh HA Manager
Waktu Eksekusi	32 detik (full migration)	≈ 10-30 detik
Data Transfer	547.1 MiB VM-state	Minimal (hanya restart)

2. Performa Jaringan

- Latensi antar node: 0.38-1.30 ms (sangat baik)
- Packet loss: 0% (koneksi stabil)
- Kecepatan transfer: 148.7 MiB/s (optimal untuk live migration)

3. Efektivitas HA Configuration

- Deteksi failure: Cepat (< 10 detik)
- Recovery time: < 30 detik
- Data integrity: Terjaga (tidak ada data loss)
- Service continuity: Minimal disruption

H. KESIMPULAN DAN REKOMENDASI

1. Kesimpulan Umum

1. Konfigurasi ZFS berhasil dengan pool dan dataset berfungsi optimal
2. Topologi jaringan mendukung live migration dan failover dengan latensi rendah
3. Live migration bekerja dengan downtime minimal (99 ms)
4. Failover berfungsi sesuai harapan dengan recovery time < 30 detik
5. HA Manager efektif dalam mendeteksi dan merespon failure

2. Rekomendasi Teknis

1. Monitoring Enhancement: Implementasi alert system untuk failover events
2. Storage Optimization: Pertimbangkan RAID-Z untuk redundancy data
3. Load Balancing: Konfigurasi HA groups untuk distribusi beban yang lebih baik
4. Backup Strategy: Schedule regular backup ke storage terpisah
5. Documentation: Update dokumentasi dengan hasil pengujian ini

3. Best Practices Terbukti

1. Shared storage (NFS) penting untuk live migration
2. Bridge network yang konsisten antar node
3. HA configuration yang tepat dengan monitor resource
4. Regular testing untuk memastikan HA tetap berfungsi

4. Untuk Pengembangan Selanjutnya

1. Stress testing dengan multiple VM concurrent migration
2. Network failure simulation untuk testing partition tolerance
3. Storage performance testing dengan beban tinggi
4. Automated testing dengan skrip untuk regular HA validation