



*Bumigora
Berbagi*

FREE ONLINE TRAINING **PROXMOX** INFRASTRUCTURE AS CODE (IAC) DENGAN **ANSIBLE**



I PUTU HARIYADI

PUTU.HARIYADI@UNIVERSITASBUMIGORA.AC.ID

www.iputuhariyadi.net

PENGENALAN VIRTUALISASI DAN PROXMOX VIRTUAL ENVIRONMENT (PVE)

APA ITU VIRTUALISASI?

- Menurut [Red Hat](#), Virtualisasi merupakan teknologi yang digunakan untuk membuat beberapa lingkungan simulasi atau sumber daya khusus dari satu sistem perangkat keras fisik.
- Perangkat lunak yang disebut **hypervisor** terhubung langsung ke perangkat keras tersebut dan membagi satu sistem tersebut menjadi lingkungan terpisah dan berbeda yang dikenal sebagai **Virtual Machine (VM)**.
- **Hypervisor** merupakan perangkat lunak yang digunakan untuk manajemen virtualisasi.
- VM mengandalkan kemampuan *hypervisor* untuk memisahkan sumber daya mesin dari perangkat keras dan mendistribusikannya dengan tepat.

JENIS VIRTUALISASI

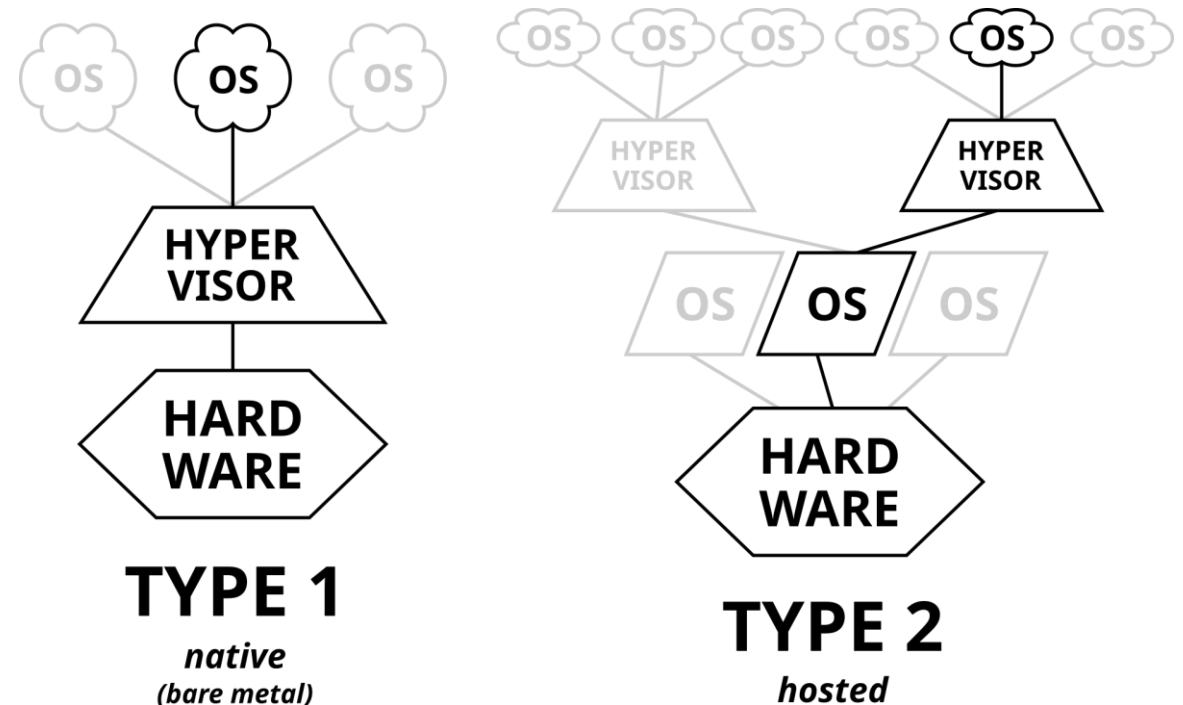
Menurut [Red Hat](#), terdapat 2 (dua) jenis **hypervisor** yaitu:

a) Type-1, native atau baremetal hypervisor.

Merupakan *hypervisor* yang berjalan secara langsung pada *hardware* dari *host* untuk mengontrol *hardware* dan manajemen *guest operating system* (OS).

b) Type-2, hosted hypervisor.

Merupakan *hypervisor* yang berjalan pada sistem operasi konvensional sebagai lapisan perangkat lunak atau aplikasi.

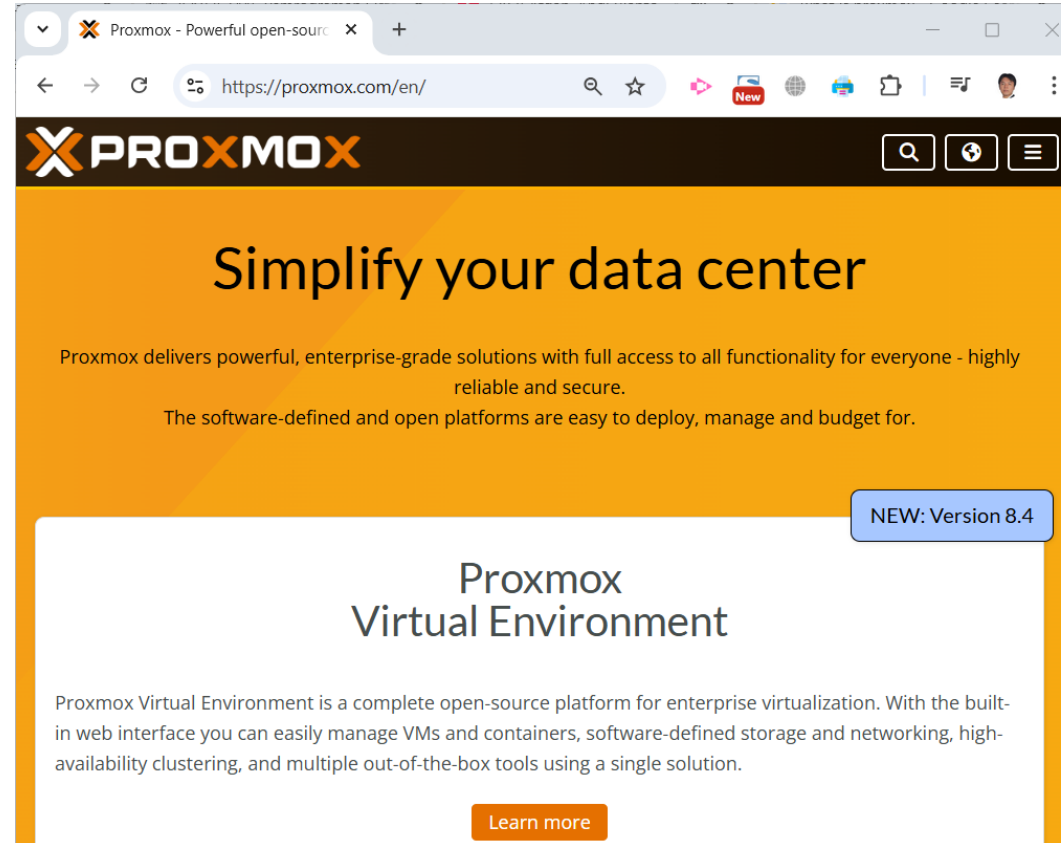


Sumber gambar: <https://en.wikipedia.org/wiki/Hypervisor#/media/File:Hyperviseur.svg>

PENGENALAN PROXMOX VIRTUAL ENVIRONMENT (PVE)

- Menurut [Wiki dari Proxmox](#), PVE merupakan platform manajemen server virtualisasi open source yang memungkinkan pengguna mengelola **Kernel-based Virtual Machine (KVM)** dan **Linux Container (LXC)**.

Informasi detail terkait
PVE dapat diakses di
<https://www.proxmox.com>



KEBUTUHAN SISTEM

Menurut situs [Proxmox](https://proxmox.com/), kebutuhan sistem untuk menjalankan PVE adalah sebagai berikut:

Recommended Hardware

- Intel 64 or AMD64 with Intel VT/AMD-V CPU flag.
- Memory, minimum 2 GB for OS and Proxmox VE services. Plus designated memory for guests. For Ceph or ZFS additional memory is required, approximately 1 GB memory for every TB used storage.
- Fast and redundant storage, best results with SSD disks.
- OS storage: Hardware RAID with batteries protected write cache ("BBU") or non-RAID with ZFS and SSD cache.
- VM storage: For local storage use a hardware RAID with battery backed write cache (BBU) or non-RAID for ZFS. Neither ZFS nor Ceph are compatible with a hardware RAID controller. Shared and distributed storage is also possible.
- Redundant Gbit NICs, additional NICs depending on the preferred storage technology and cluster setup – 10 Gbit and higher is also supported.
- For PCI(e) passthrough a CPU with VT-d/AMD-d CPU flag is needed.

KEBUTUHAN SISTEM

For Evaluation

Minimum Hardware (for testing only)

- CPU: 64bit (Intel 64 or AMD64)
- Intel VT/AMD-V capable CPU/Mainboard (for KVM full virtualization support)
- Minimum 1 GB RAM
- Hard drive
- One NIC

See more details in the [Requirements documentation](#)

Testing with desktop virtualization

Proxmox VE can be installed as a guest on all common used desktop virtualization solutions as long as they support nested virtualization.

KEBUTUHAN SISTEM

Supported web browsers for accessing the web interface

To use the web interface you need a modern browser, this includes:

- Firefox, a release from the current year, or the latest Extended Support Release
- Chrome, a release from the current year
- Microsofts currently supported version of Edge
- Safari, a release from the current year

Installation on a Debian system

If you want you can install Proxmox VE on top of a running Debian 64-bit. This is especially interesting, if you want to have a custom partition layout.



INSTALASI DAN KONFIGURASI DASAR PVE DI VIRTUAL PRIVATE SERVER (VPS)

SPEKIFIKASI VPS YANG DISEWA DI IDCLOUDHOST UNTUK INSTALASI PVE

- Sistem Operasi Debian 12.
- 2 CPU.
- 4 GB RAM.
- 60 GB DISK.

The screenshot shows the 'Create New Resource' interface on the IDCloudHost dashboard. The interface is divided into several sections for configuring a new Virtual Machine:

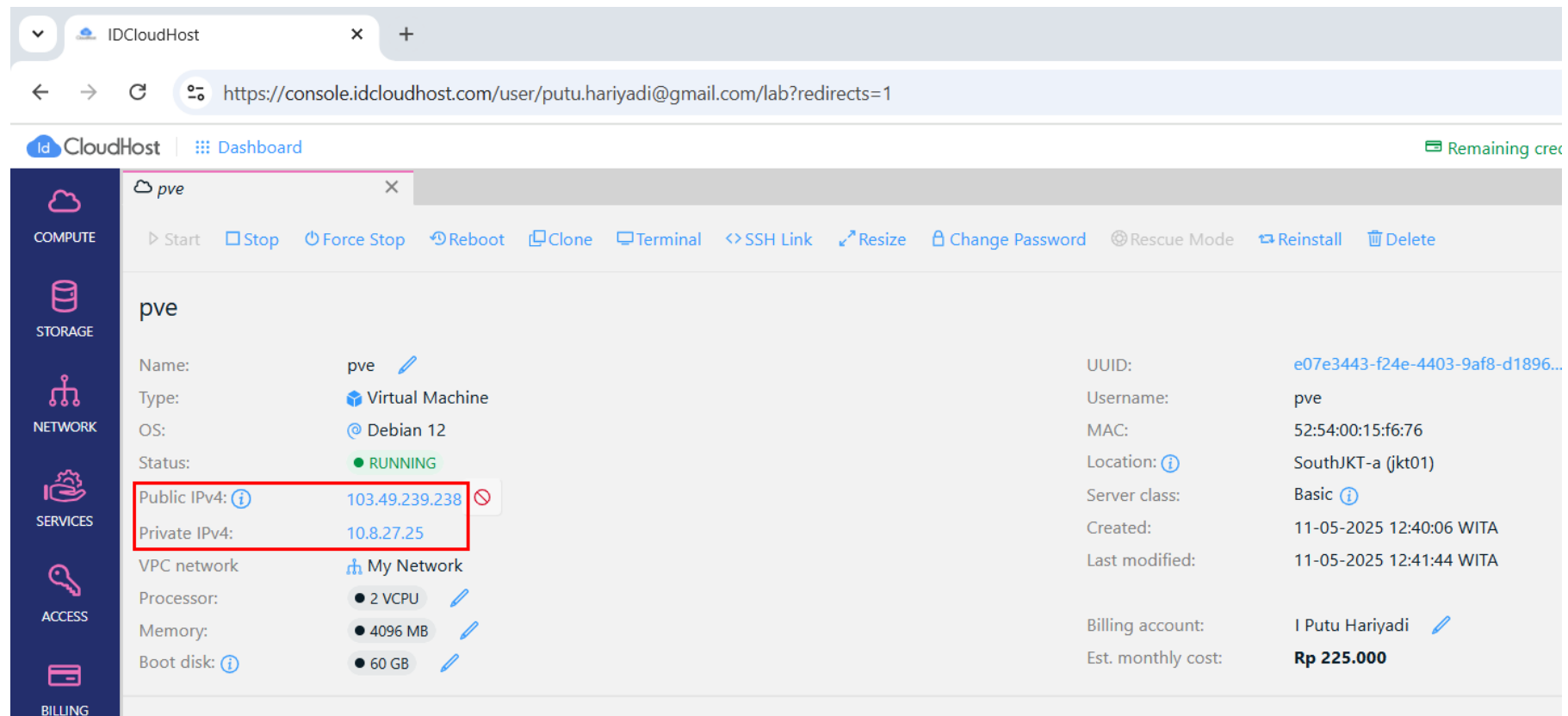
- Type*:** Options for 'Virtual Machine' (selected) and 'App Catalog'.
- OS*:** A grid of operating system choices including Ubuntu, Debian (selected), Almalinux, Centos, Fedora, Opensuse, CloudLinux, Rocky Linux, VzLinux, RHEL, and Windows. Each option includes a version dropdown menu.
- Location*:** Options for 'Indonesia' (selected) and 'Singapore', with a 'SouthJKT-a' location dropdown.
- Server class*:** Options for 'Basic Standard', 'Intel eXtreme 5x Faster', and 'AMD eXtreme 6x Faster'.
- Size*:** A table of resource packages with their respective prices and specifications.

Package	Price	CPU	RAM	Disk
Basic Standard	Rp 55.000 / month Rp 68 / hour	1 CPU	1 GB RAM	20 GB DISK
2 CPU	Rp 87.000 / month Rp 120 / hour	2 CPU	2 GB RAM	20 GB DISK
Intel eXtreme	Rp 100.000 / month Rp 137 / hour	2 CPU	2 GB RAM	40 GB DISK
AMD eXtreme	Rp 225.000 / month Rp 308 / hour	2 CPU	4 GB RAM	60 GB DISK

At the bottom right, there are sliders for customizing CPU (2 to 32), GB RAM (2 to 16), and GB DISK (20 to 1000).

VPS IDCLOUDHOST UNTUK INSTALASI PVE

VPS yang disewa di [IDCloudHost](https://idcloudhost.com) memiliki 2 (dua) alamat **IPv4** yaitu satu **IP Publik**, sebagai contoh **103.49.239.238** dan satu **IP Private**, sebagai contoh **10.8.27.25**.

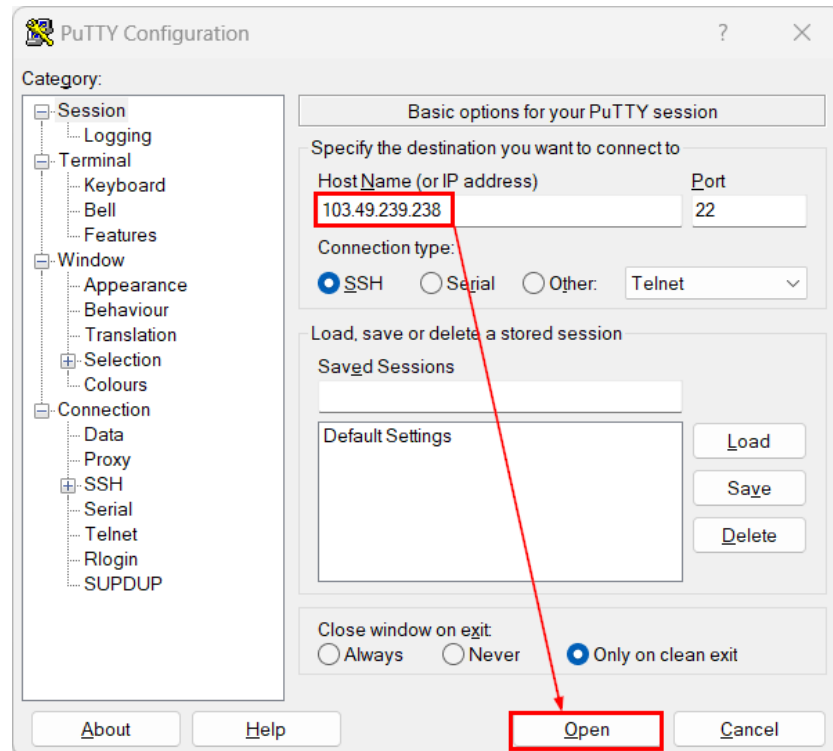


The screenshot displays the IDCloudHost console interface. The browser address bar shows the URL: `https://console.idcloudhost.com/user/putu.hariyadi@gmail.com/lab?redirects=1`. The console header includes the IDCloudHost logo, a "Dashboard" link, and a "Remaining credit" indicator. A sidebar on the left contains navigation icons for COMPUTE, STORAGE, NETWORK, SERVICES, ACCESS, and BILLING. The main content area shows the details of a virtual machine named "pve". At the top, there is a row of action buttons: Start, Stop, Force Stop, Reboot, Clone, Terminal, SSH Link, Resize, Change Password, Rescue Mode, Reinstall, and Delete. Below this, the VM details are organized into two columns. The left column lists: Name (pve), Type (Virtual Machine), OS (Debian 12), Status (RUNNING), Public IPv4 (103.49.239.238), Private IPv4 (10.8.27.25), VPC network (My Network), Processor (2 VCPU), Memory (4096 MB), and Boot disk (60 GB). The right column lists: UUID (e07e3443-f24e-4403-9af8-d1896...), Username (pve), MAC (52:54:00:15:f6:76), Location (SouthJKT-a (jkt01)), Server class (Basic), Created (11-05-2025 12:40:06 WITA), Last modified (11-05-2025 12:41:44 WITA), Billing account (I Putu Hariyadi), and Est. monthly cost (Rp 225.000). A red rectangle highlights the Public and Private IPv4 addresses.

Field	Value
Name	pve
Type	Virtual Machine
OS	Debian 12
Status	RUNNING
Public IPv4	103.49.239.238
Private IPv4	10.8.27.25
VPC network	My Network
Processor	2 VCPU
Memory	4096 MB
Boot disk	60 GB
UUID	e07e3443-f24e-4403-9af8-d1896...
Username	pve
MAC	52:54:00:15:f6:76
Location	SouthJKT-a (jkt01)
Server class	Basic
Created	11-05-2025 12:40:06 WITA
Last modified	11-05-2025 12:41:44 WITA
Billing account	I Putu Hariyadi
Est. monthly cost	Rp 225.000

MENGAKSES VPS MENGGUNAKAN PUTTY SSH CLIENT

1. Mengunduh file binary **Putty SSH Client 64 bit x86** pada alamat <https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe>
2. Menjalankan aplikasi **Putty** yang telah diunduh.



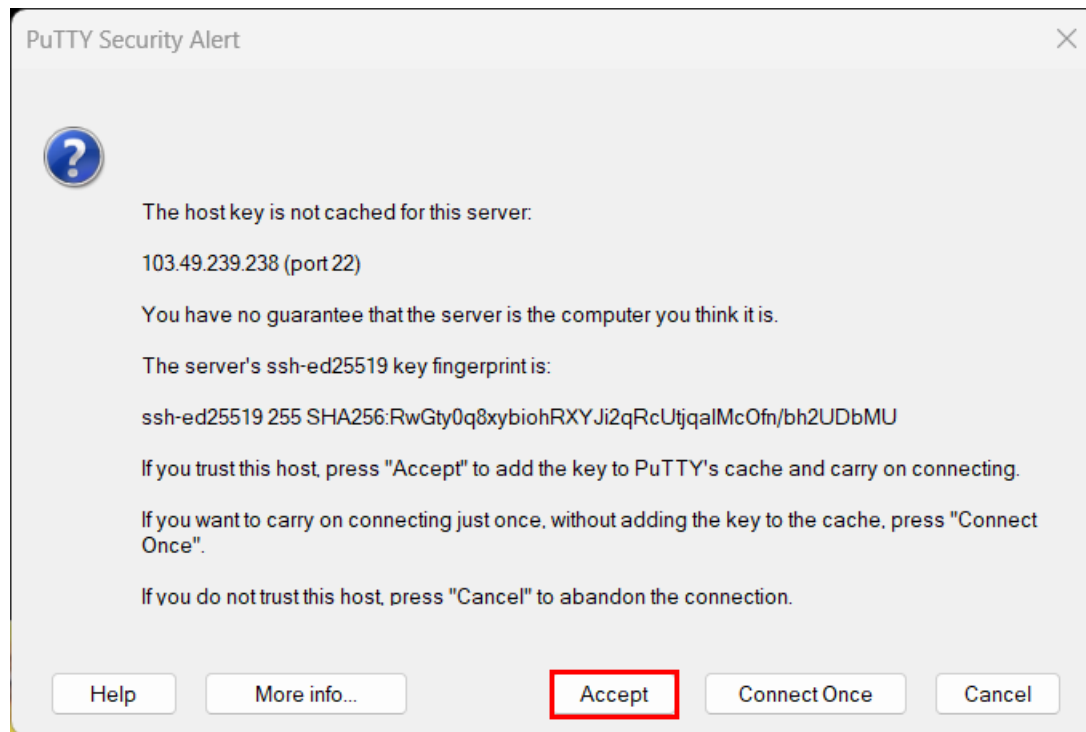
Tampil kotak dialog **Putty Configuration**.

Pada isian **Host Name (or IP Address)**, masukkan alamat **IP Publik** dari **VPS**, sebagai contoh **103.49.239.238**. **Sesuaikan alamat IP Publik tersebut dengan alamat dari VPS yang dimiliki.**

Klik tombol **Open** untuk melakukan koneksi SSH ke VPS.

MENGAKSES VPS MENGGUNAKAN PUTTY SSH CLIENT

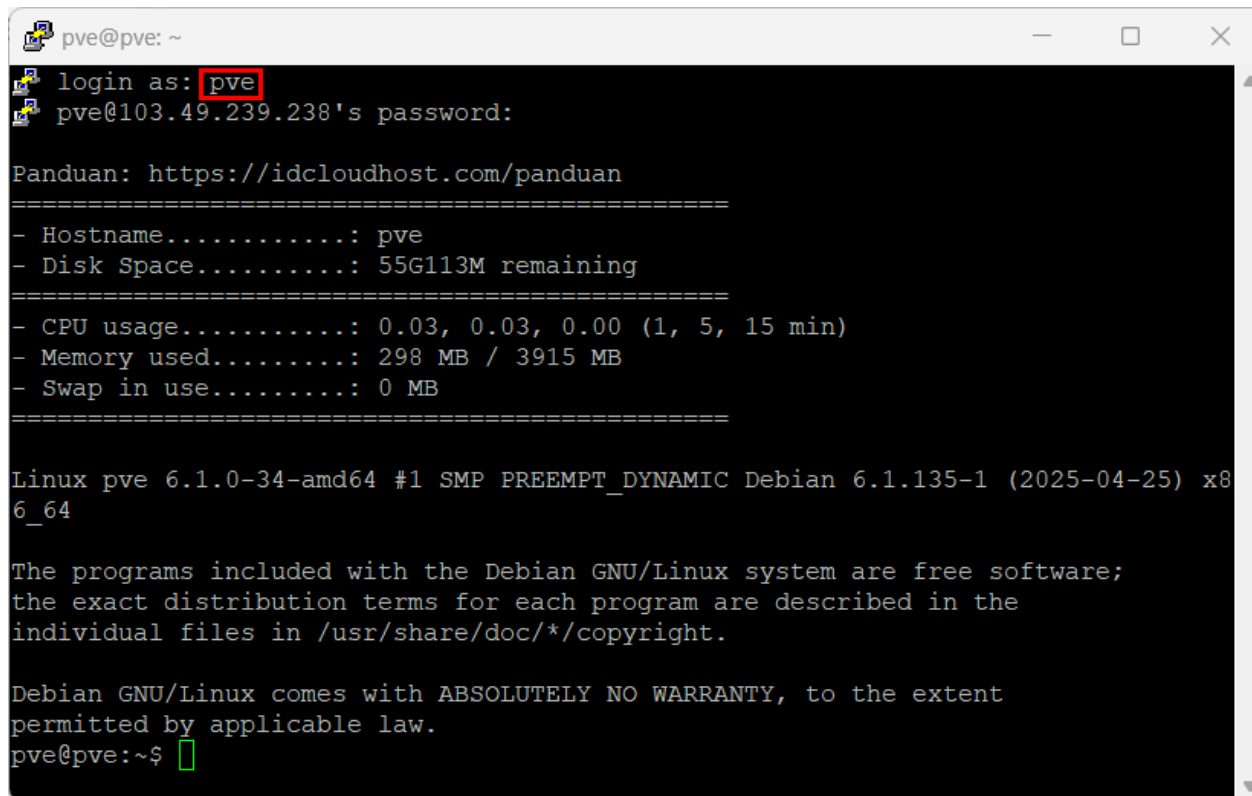
Tampil kotak dialog **PuTTY Security Alert** yang menampilkan pesan peringatan terkait potensi pelanggaran keamanan.



Klik tombol **Accept** untuk melanjutkan.

MENGAKSES VPS MENGGUNAKAN PUTTY SSH CLIENT

Tampil kotak dialog **Putty** yang meminta pengguna untuk melakukan proses otentikasi login ke VPS.



```
pve@pve: ~
login as: pve
pve@103.49.239.238's password:

Panduan: https://idcloudhost.com/panduan
=====
- Hostname.....: pve
- Disk Space.....: 55G113M remaining
=====
- CPU usage.....: 0.03, 0.03, 0.00 (1, 5, 15 min)
- Memory used.....: 298 MB / 3915 MB
- Swap in use.....: 0 MB
=====

Linux pve 6.1.0-34-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.135-1 (2025-04-25) 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.
pve@pve:~$
```

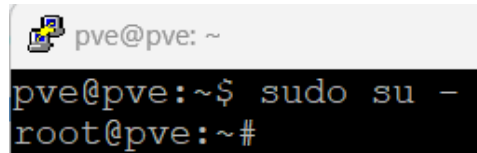
Pada inputan **login as:**, masukkan nama *login*, sebagai contoh “**pve**” dan tekan tombol **Enter**. **Sesuaikan nama login dengan yang dimiliki pada VPS**. Selanjutnya tampil inputan **password:**, masukkan sandi *login* pengguna dan tekan tombol **Enter**.

Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt* \$.

MENGATUR HOSTNAME AGAR DAPAT DIPETAKAN KE ALAMAT IP PRIVATE PADA FILE /ETC/HOSTS

1. Berpindah *user* sebagai “**root**”.

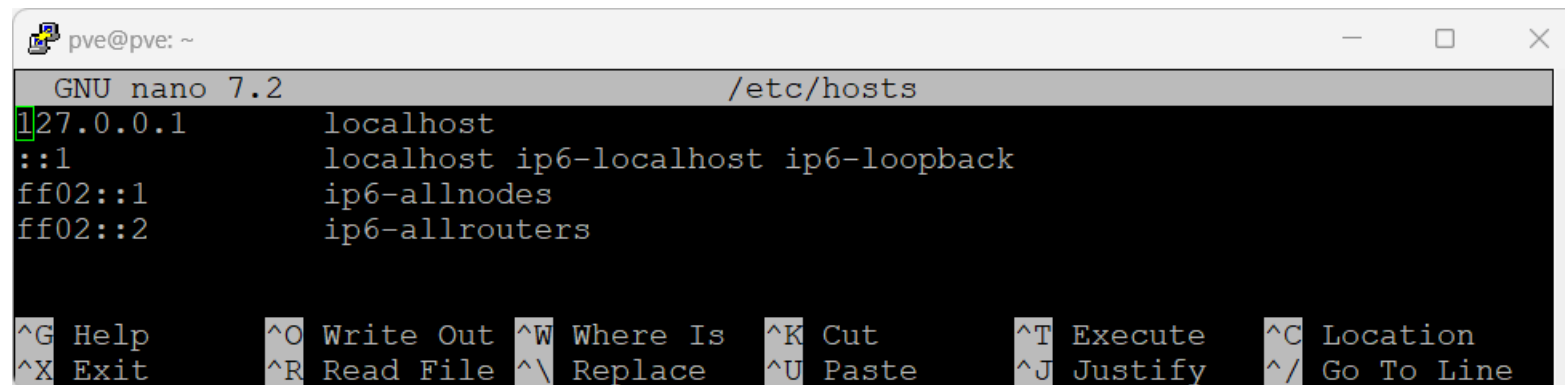
`sudo su -`



```
pve@pve: ~  
pve@pve:~$ sudo su -  
root@pve:~#
```

2. Membuka *file* **/etc/hosts** menggunakan editor **nano**.

`nano /etc/hosts`



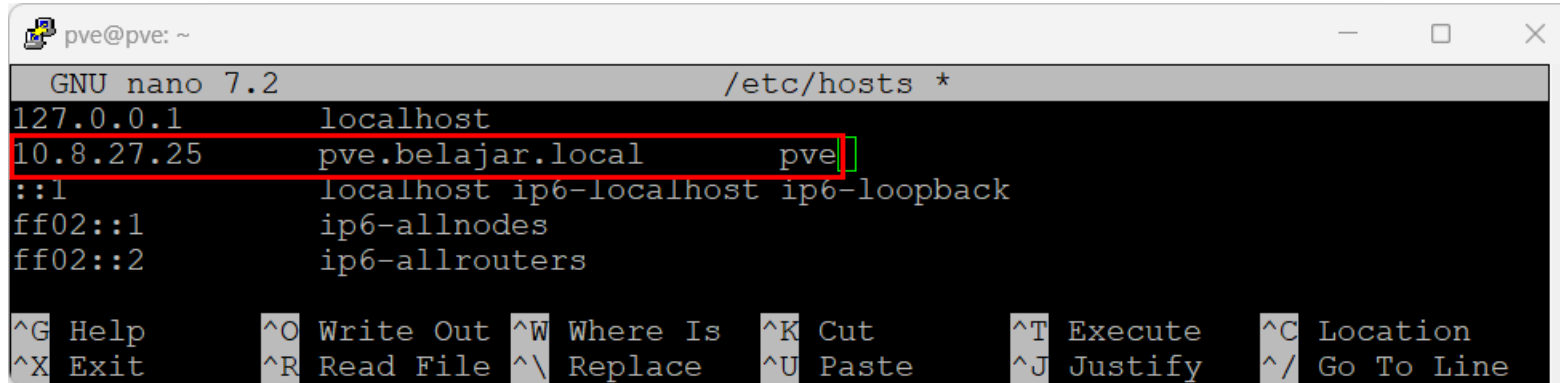
```
pve@pve: ~  
GNU nano 7.2 /etc/hosts  
1 127.0.0.1 localhost  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

MENGATUR HOSTNAME AGAR DAPAT DIPETAKAN KE ALAMAT IP PRIVATE PADA FILE /ETC/HOSTS

Pada baris kedua tambahkan pengaturan pemetaan alamat **IP Private** ke **Hostname** dengan format penulisan sebagai berikut:

IPPrivate Hostname Alias

Sehingga hasilnya terlihat seperti berikut:



```
pve@pve: ~  
GNU nano 7.2 /etc/hosts *  
127.0.0.1 localhost  
10.8.27.25 pve.belajar.local pve  
::1 localhost ip6-localhost ip6-loopback  
ff02::1 ip6-allnodes  
ff02::2 ip6-allrouters  
^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
```

Sesuaikan **IP Private**, **Hostname** dalam **Fully Qualified Domain Name (FQDN)** dan **Alias** dengan nilai yang dimiliki atau digunakan VPS.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

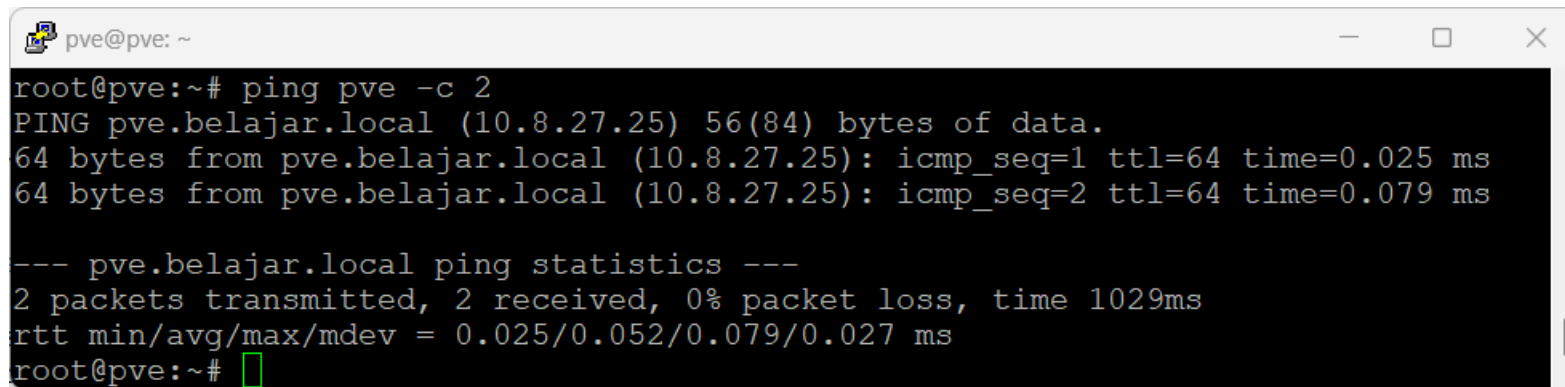
MENGATUR HOSTNAME AGAR DAPAT DIPETAKAN KE ALAMAT IP PRIVATE PADA FILE /ETC/HOSTS

3. Memverifikasi hasil pengaturan.

```
hostname --ip-address
```

```
root@pve:~# hostname --ip-address  
10.8.27.25
```

```
ping pve -c 2
```

A terminal window titled 'pve@pve: ~' showing the execution of the 'ping pve -c 2' command. The output displays two successful ping responses from 'pve.belajar.local' (10.8.27.25) with ICMP sequence numbers 1 and 2, and response times of 0.025 ms and 0.079 ms respectively. Below the responses, it shows ping statistics: 2 packets transmitted, 2 received, 0% packet loss, and a total time of 1029ms. The round-trip times (rtt) are listed as min/avg/max/mdev = 0.025/0.052/0.079/0.027 ms. The prompt 'root@pve:~#' is followed by a green cursor.

```
pve@pve: ~  
root@pve:~# ping pve -c 2  
PING pve.belajar.local (10.8.27.25) 56(84) bytes of data.  
64 bytes from pve.belajar.local (10.8.27.25): icmp_seq=1 ttl=64 time=0.025 ms  
64 bytes from pve.belajar.local (10.8.27.25): icmp_seq=2 ttl=64 time=0.079 ms  
  
--- pve.belajar.local ping statistics ---  
2 packets transmitted, 2 received, 0% packet loss, time 1029ms  
rtt min/avg/max/mdev = 0.025/0.052/0.079/0.027 ms  
root@pve:~#
```

Sesuaikan *alias* **pve** dengan yang dimiliki atau digunakan VPS.

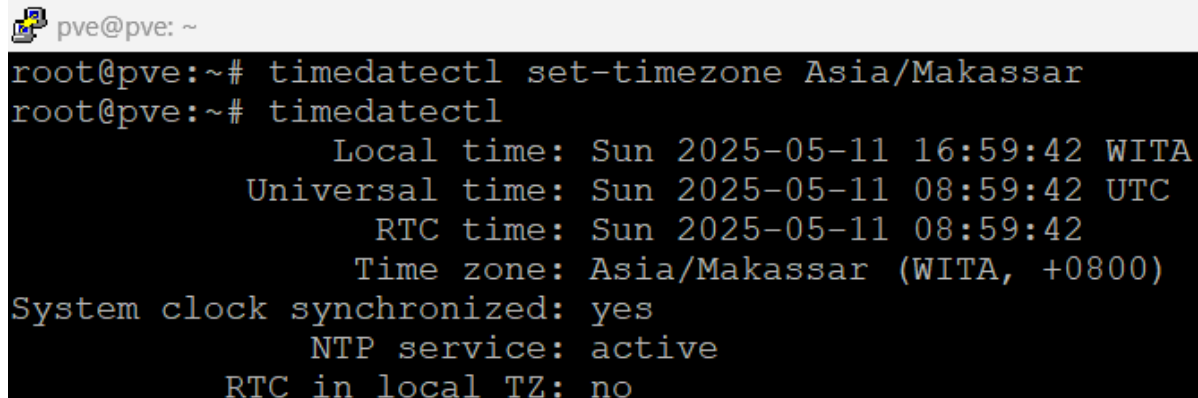
MENGATUR TIMEZONE

Gunakan **Asia/Jakarta** jika berada pada zona **Waktu Indonesia Barat (WIB)**, **Asia/Makassar** jika berada pada zona **Waktu Indonesia Tengah (WITA)** dan **Asia/Jayapura** jika berada pada zona **Waktu Indonesia Timur (WIT)**.

```
timedatectl set-timezone Asia/Makassar
```

Memverifikasi hasil pengaturan dengan mengeksekusi perintah

```
timedatectl
```



```
pve@pve: ~  
root@pve:~# timedatectl set-timezone Asia/Makassar  
root@pve:~# timedatectl  
          Local time: Sun 2025-05-11 16:59:42 WITA  
    Universal time: Sun 2025-05-11 08:59:42 UTC  
           RTC time: Sun 2025-05-11 08:59:42  
        Time zone: Asia/Makassar (WITA, +0800)  
System clock synchronized: yes  
          NTP service: active  
    RTC in local TZ: no
```

MENAMBAHKAN REPOSITORY PVE

```
echo "deb [arch=amd64] http://download.proxmox.com/debian/pve bookworm pve-no-subscription" \  
> /etc/apt/sources.list.d/pve-install-repo.list
```

```
root@pve:~# echo "deb [arch=amd64] http://download.proxmox.com/debian/pve bookworm pve-no-subscription" \  
> /etc/apt/sources.list.d/pve-install-repo.list
```

Memverifikasi hasil penambahan repository PVE

```
cat /etc/apt/sources.list.d/pve-install-repo.list
```

```
root@pve:~# cat /etc/apt/sources.list.d/pve-install-repo.list  
deb [arch=amd64] http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

MENAMBAHKAN PVE REPOSITORY KEY

```
wget https://enterprise.proxmox.com/debian/proxmox-release-bookworm.gpg \
-O /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

```
root@pve:~# wget https://enterprise.proxmox.com/debian/proxmox-release-bookworm.gpg \
-O /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
--2025-05-11 17:16:04-- https://enterprise.proxmox.com/debian/proxmox-release-bookworm.gpg
Resolving enterprise.proxmox.com (enterprise.proxmox.com)... 117.120.5.24, 2407:6180:1:a:41::50
Connecting to enterprise.proxmox.com (enterprise.proxmox.com)|117.120.5.24|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1187 (1.2K) [application/octet-stream]
Saving to: '/etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg'

/etc/apt/trusted.gpg.d/proxmox-release- 100%[=====>] 1.16K --.-KB/s in 0s
2025-05-11 17:16:06 (17.9 MB/s) - '/etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg' saved [1187/1187]
```

Memverifikasi PVE Repository Key

```
sha512sum /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
```

```
root@pve:~# sha512sum /etc/apt/trusted.gpg.d/proxmox-release-bookworm.gpg
7da6fe34168adc6e479327ba517796d4702fa2f8b4f0a9833f5ea6e6b48f6507a6da403a274fe201
595edc86a84463d50383d07f64bdde2e3658108db7d6dc87 /etc/apt/trusted.gpg.d/proxmox
-release-bookworm.gpg
```

MEMPERBARUI REPOSITORY DAN SISTEM

`apt update && apt full-upgrade -y`

```
root@pve:~# apt update && apt full-upgrade -y
Get:1 file:/etc/apt/mirrors/debian.list Mirrorlist [30 B]
Get:2 file:/etc/apt/mirrors/debian-security.list Mirrorlist [39 B]
Hit:3 https://deb.debian.org/debian bookworm InRelease
Hit:4 https://deb.debian.org/debian bookworm-updates InRelease
Hit:5 https://deb.debian.org/debian bookworm-backports InRelease
Hit:6 https://deb.debian.org/debian-security bookworm-security InRelease
Hit:7 http://download.proxmox.com/debian/pve bookworm InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
  grub-common grub-efi-amd64-bin grub-efi-amd64-signed grub-pc-bin
  grub2-common shim-helpers-amd64-signed shim-signed shim-signed-common
  shim-unsigned
9 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

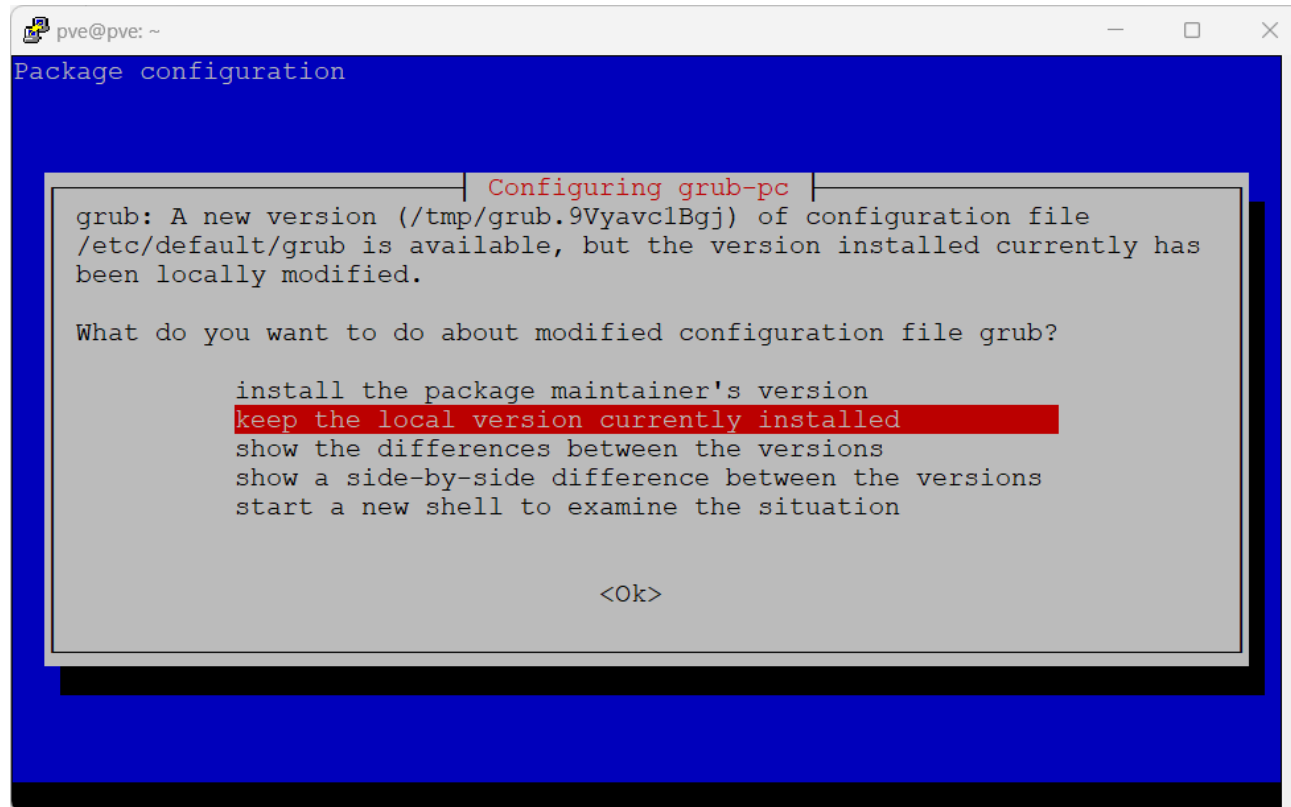
MENGINSTALASI PVE KERNEL

```
apt install proxmox-default-kernel -y
```

```
root@pve:~# apt install proxmox-default-kernel -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  busybox grub-pc proxmox-kernel-6.8 proxmox-kernel-6.8.12-10-pve-signed
  pve-firmware
Suggested packages:
  linux-image
The following packages will be REMOVED:
  firmware-linux-free
The following NEW packages will be installed:
  busybox grub-pc proxmox-default-kernel proxmox-kernel-6.8
  proxmox-kernel-6.8.12-10-pve-signed pve-firmware
0 upgraded, 6 newly installed, 1 to remove and 0 not upgraded.
```

MENGINSTALASI PVE KERNEL

Tampil kotak dialog **Configuring grub-pc** untuk konfigurasi **GRUB**. Secara *default* terpilih **Keep the local version currently installed** untuk menyimpan versi lokal yang telah terinstalasi.



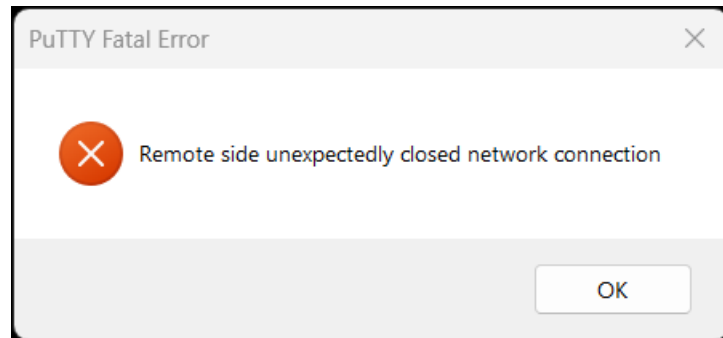
Tekan **Enter** untuk melanjutkan proses instalasi.

MENGINSTALASI PVE KERNEL

Melakukan *restart* VPS.

```
systemctl reboot
```

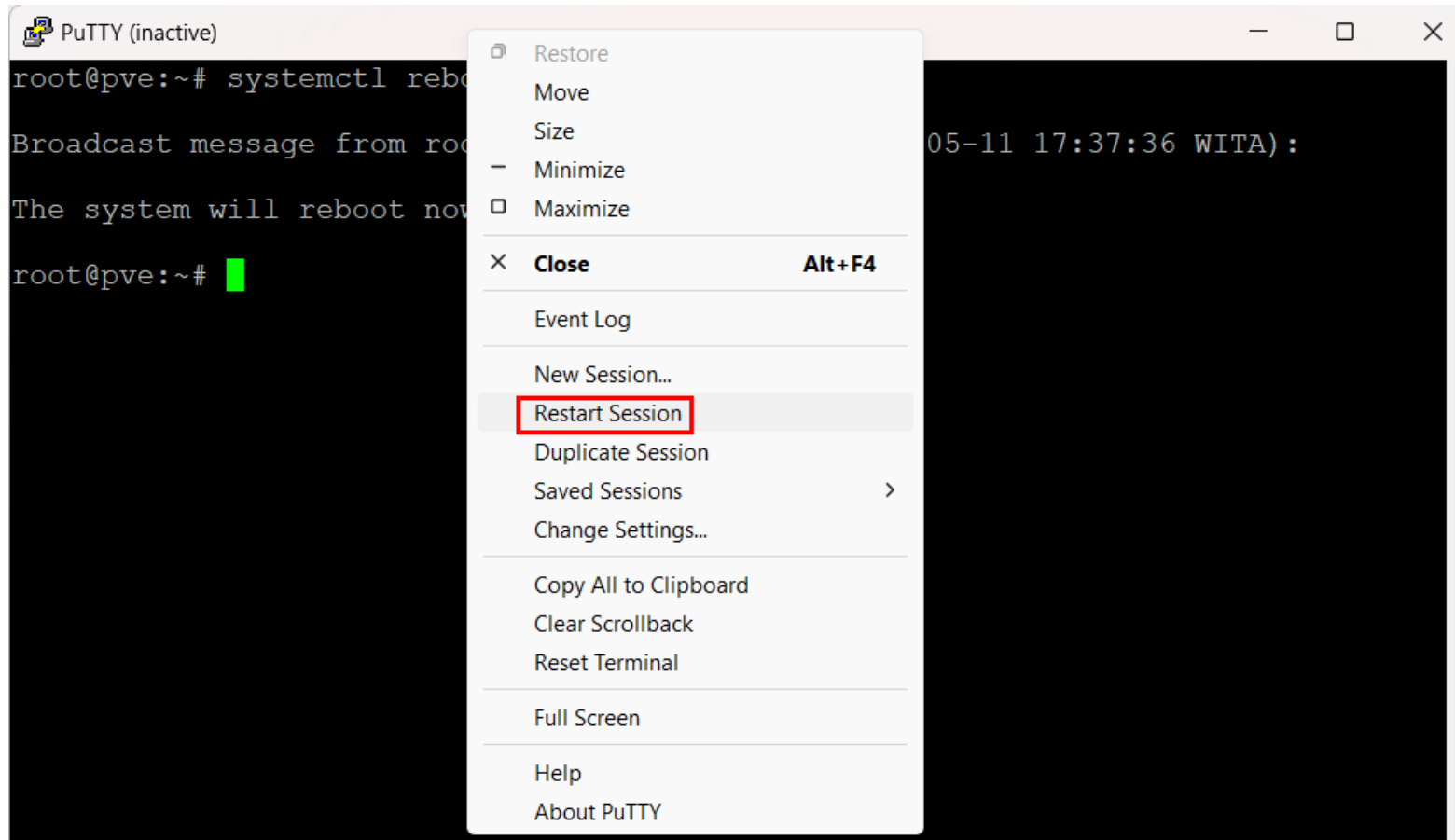
Tampil kotak dialog **Putty Fatal Error**.



Tekan tombol **OK** untuk menutup kotak dialog tersebut.

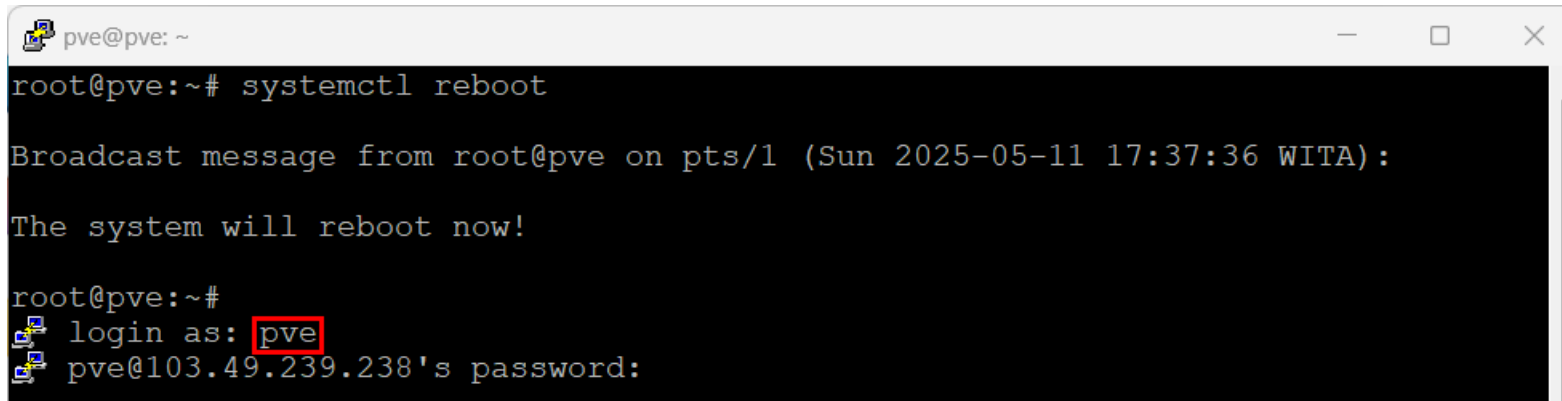
MENGINSTALASI PVE KERNEL

Tunggu beberapa saat dan lakukan koneksi SSH kembali ke VPS dengan cara klik kanan pada **title bar** dari **PuTTY (inactive)** dan memilih **Restart Session**.



MENGINSTALASI PVE KERNEL

Tampil inputan **login as:** yang meminta pengguna untuk melakukan proses otentikasi login ke VPS.

A terminal window titled 'pve@pve: ~' with standard window controls. The terminal shows the command 'systemctl reboot' being executed as root. It displays a broadcast message indicating a reboot at 17:37:36 WITA. After the reboot, the prompt returns to root@pve. The user then enters 'login as: pve', where 'pve' is highlighted with a red box. The next prompt is 'pve@103.49.239.238's password:'.

```
pve@pve: ~  
root@pve:~# systemctl reboot  
  
Broadcast message from root@pve on pts/1 (Sun 2025-05-11 17:37:36 WITA):  
  
The system will reboot now!  
  
root@pve:~#  
login as: pve  
pve@103.49.239.238's password:
```

Pada masukkan nama *login*, sebagai contoh “**pve**” dan tekan tombol **Enter**.
Sesuaikan nama login dengan yang dimiliki pada VPS. Selanjutnya tampil inputan **password:**, masukkan sandi *login* pengguna dan tekan tombol **Enter**.

Apabila proses otentikasi berhasil dilakukan maka akan tampil *shell prompt* \$.

MENGINSTALASI PVE PACKAGES

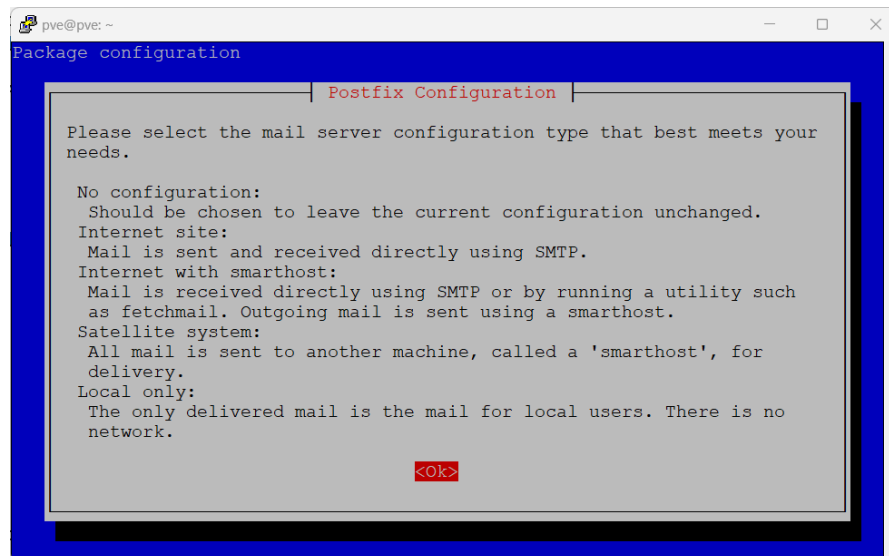
1. Berpindah *user* sebagai “**root**”.

`sudo su -`

2. Menginstalasi paket-paket PVE

`apt install proxmox-ve postfix open-iscsi chrony`

Tekan **y** untuk melanjutkan instalasi.

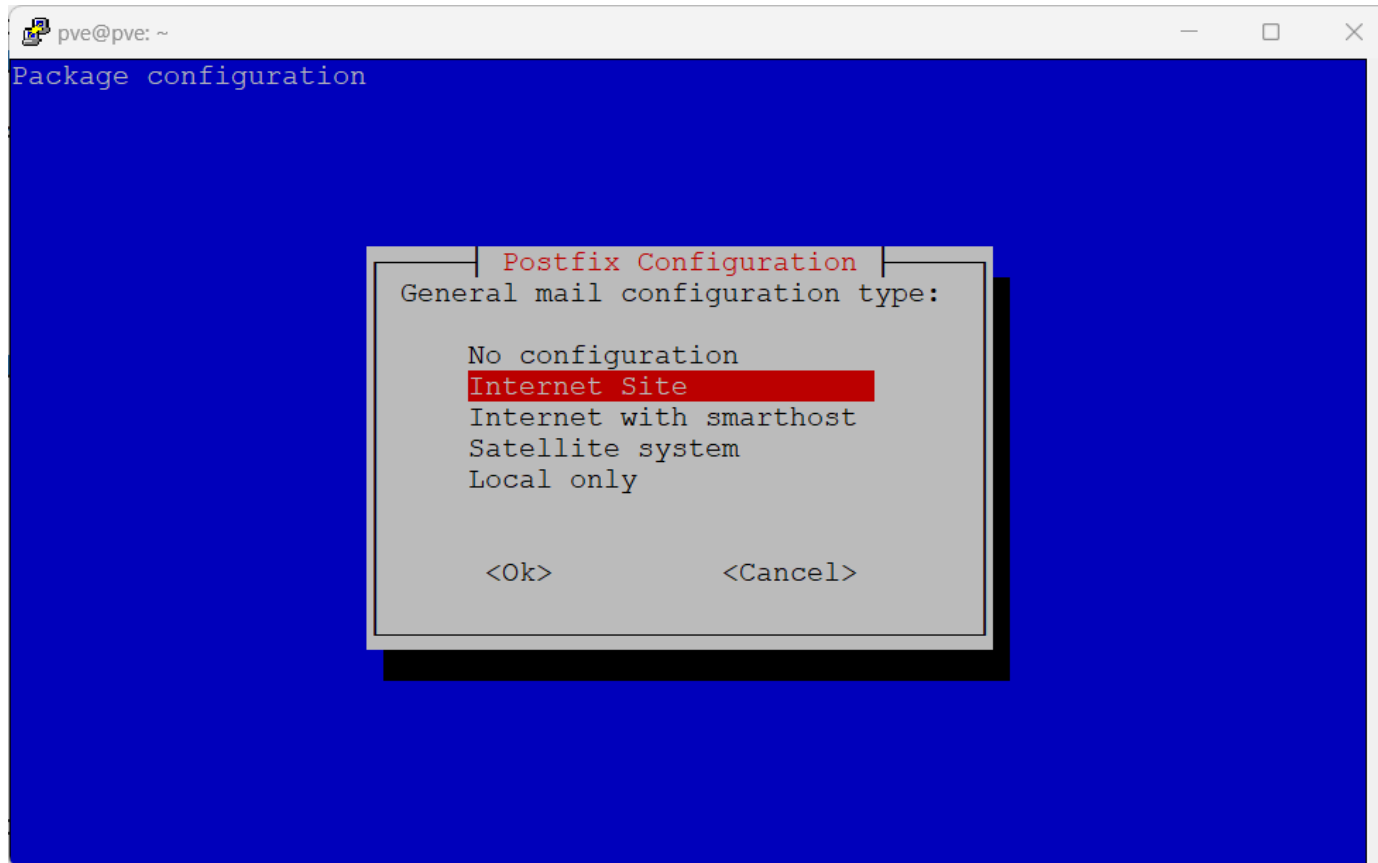


Tampil kotak dialog **Postfix Configuration** untuk memilih tipe konfigurasi server mail.

Tekan **Enter** untuk melanjutkan.

MENGINSTALASI PVE PACKAGES

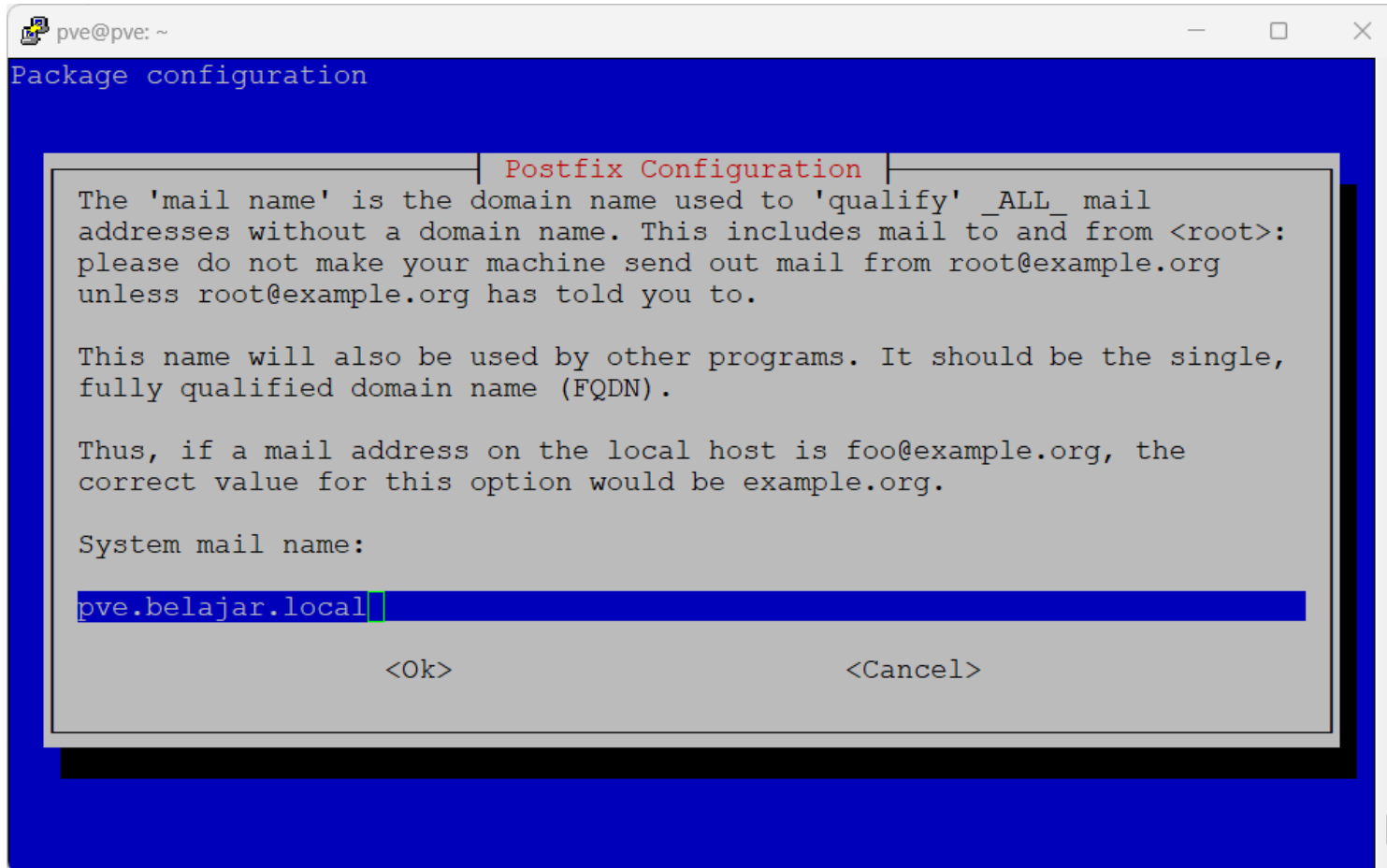
Tampil kotak dialog **Postfix Configuration** untuk menentukan tipe konfigurasi mail.



Secara *default* terpilih **Internet Site**.
Tekan **Enter** untuk melanjutkan.

MENGINSTALASI PVE PACKAGES

Tampil kotak dialog **Postfix Configuration** untuk mengatur **System mail name**.



Secara *default* pada inputan **System mail name** telah tertera **pve.belajar.local**.

Tekan **Enter** untuk melanjutkan.

Tunggu hingga keseluruhan proses instalasi paket-paket tersebut selesai dilakukan.

MENGHAPUS KERNEL DEBIAN

- Penghapusan ini dilakukan karena PVE membawa kernelnya sendiri.

```
apt -y remove linux-image-amd64 'linux-image-6.1*'
```

- Memperbarui dan mengecek konfigurasi **Grub2**.

```
update-grub
```

MENGHAPUS PAKET OS-PROBER*

```
apt -y remove os-prober
```

Paket **os-prober** melakukan pemindaian (*scan*) seluruh partisi dari *host* untuk membuat entri *dual-boot* di **GRUB**.

* jika paket tersebut terinstalasi pada sistem

MENGATUR PASSWORD USER ROOT

Pengaturan *password* untuk *user root* dilakukan agar *user* tersebut dapat digunakan login ke **Web Graphical User Interface (GUI)** dari PVE.

`passwd`

```
root@pve:~# passwd
New password:
Retype new password:
passwd: password updated successfully
```

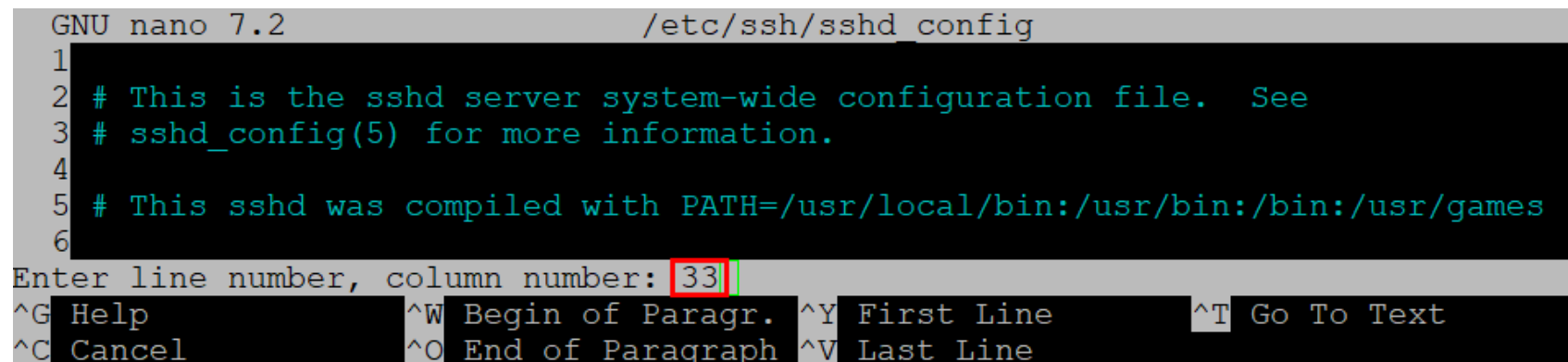
Pada inputan **New password:** dan **Retype new password:** masukkan sandi yang digunakan oleh *user root*.

MENGIJINKAN USER ROOT LOGIN VIA SECURE SHELL (SSH)

Mengatur agar user “**root**” dapat digunakan *login* ke **SSH Server** dari PVE dengan menambahkan *directive* **PermitRootLogin yes** pada file `/etc/ssh/sshd_config` menggunakan editor **nano**.

```
nano -l /etc/ssh/sshd_config
```

Berpindah ke baris 33 yang memuat parameter atau *directive* **PermitRootLogin**.



```
GNU nano 7.2 /etc/ssh/sshd_config
1
2 # This is the sshd server system-wide configuration file.  See
3 # sshd_config(5) for more information.
4
5 # This sshd was compiled with PATH=/usr/local/bin:/usr/bin:/bin:/usr/games
6
Enter line number, column number: 33
^G Help      ^W Begin of Paragr. ^Y First Line  ^T Go To Text
^C Cancel    ^O End of Paragraph ^V Last Line
```

Tekan **CTRL+_** dan masukkan nomor baris **33** pada inputan **Enter line number, column number:** serta tekan **Enter**.

MENGIJINKAN USER ROOT LOGIN VIA SECURE SHELL (SSH)

Tampil baris 33 yang memuat nilai **#PermitRootLogin prohibit-password**.

```
GNU nano 7.2 /etc/ssh/sshd config
30 # Authentication:
31
32 #LoginGraceTime 2m
33 #PermitRootLogin prohibit-password
34 #StrictModes yes
```

Hilangkan tanda **#** di awal baris **33** tersebut dan ubah nilai parameter **PermitRootLogin** dari **prohibit-password** menjadi **yes** sehingga terlihat seperti berikut:

```
GNU nano 7.2 /etc/ssh/sshd config *
30 # Authentication:
31
32 #LoginGraceTime 2m
33 PermitRootLogin yes
34 #StrictModes yes
```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MENGIJINKAN USER ROOT LOGIN VIA SECURE SHELL (SSH)

Melakukan **restart service sshd** agar perubahan konfigurasi berdampak.

```
systemctl restart sshd
```

Memverifikasi status dari *service sshd*.

```
systemctl status sshd
```

```
root@pve:~# systemctl status sshd
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Mon 2025-05-12 20:48:09 WITA; 34s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 4727 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 4728 (sshd)
    Tasks: 1 (limit: 4632)
   Memory: 1.4M
      CPU: 30ms
   CGroup: /system.slice/ssh.service
           └─4728 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

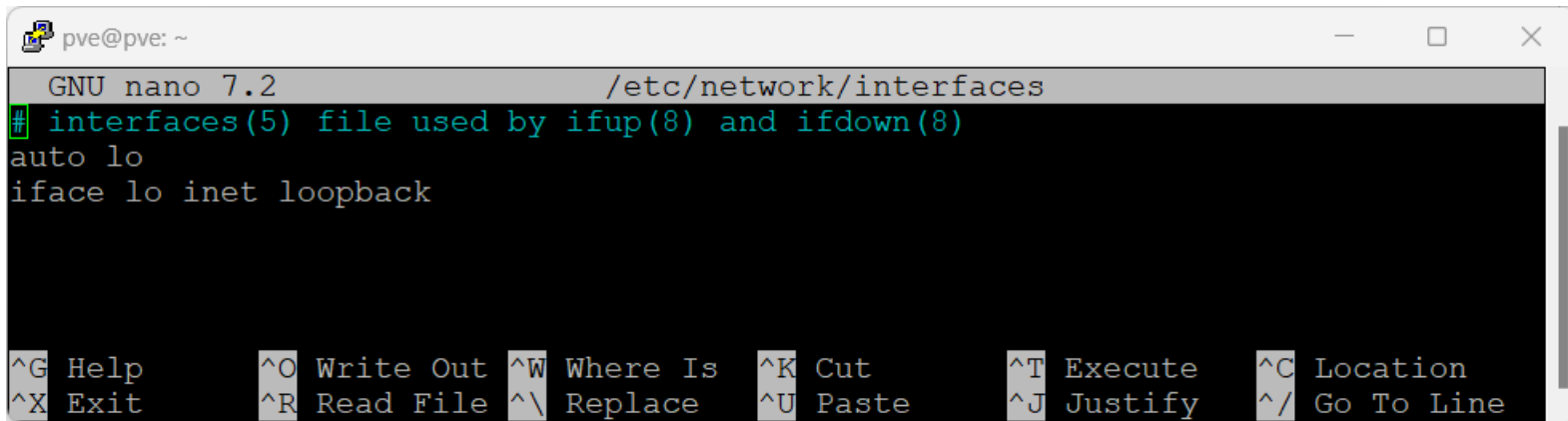
May 12 20:48:09 pve sshd[4728]: Server listening on 0.0.0.0 port 22.
May 12 20:48:09 pve systemd[1]: Stopping ssh.service - OpenBSD Secure Shell ser
May 12 20:48:09 pve sshd[4728]: Server listening on :: port 22.
May 12 20:48:09 pve systemd[1]: ssh.service: Deactivated successfully.
May 12 20:48:09 pve systemd[1]: Stopped ssh.service - OpenBSD Secure Shell serv
May 12 20:48:09 pve systemd[1]: Starting ssh.service - OpenBSD Secure Shell ser
May 12 20:48:09 pve systemd[1]: Started ssh.service - OpenBSD Secure Shell serv
lines 1-20/20 (END)
```

Tekan tombol **q** untuk keluar dari *output* status *sshd* tersebut.

MENGATUR INTERFACE BRIDGE VMBRO PADA FILE /ETC/NETWORK/INTERFACES

Penambahan *interface bridge vmbro* pada *file* konfigurasi **/etc/network/interfaces** dilakukan agar **Virtual Machine (VM)/Container (CT)** dapat terhubung ke jaringan sehingga dapat saling berkomunikasi satu dengan lainnya di jaringan serta terkoneksi ke *Internet*.

```
nano /etc/network/interfaces
```



```
pve@pve: ~  
GNU nano 7.2 /etc/network/interfaces  
# interfaces(5) file used by ifup(8) and ifdown(8)  
auto lo  
iface lo inet loopback  
  
^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
```

MENGATUR INTERFACE BRIDGE VMBR0 PADA FILE /ETC/NETWORK/INTERFACES

Tambahkan entri berikut pada bagian akhir dari *file* **/etc/network/interfaces** untuk membuat *interface bridge vubr0* dengan alamat IP yang diatur secara statik yaitu **192.168.0.1/24**.

```
iface ens3 inet manual
```

```
auto vubr0
```

```
iface vubr0 inet static
```

```
    address 192.168.0.1/24
```

```
    bridge-ports none
```

```
    bridge-stp off
```

```
    bridge-fd 0
```

MENGATUR INTERFACE BRIDGE VMBRO PADA FILE /ETC/NETWORK/INTERFACES

Hasil penambahan akan terlihat seperti berikut:

```
GNU nano 7.2 /etc/network/interfaces
# network interface settings; autogenerated
# Please do NOT modify this file directly, unless you know what
# you're doing.
#
# If you want to manage parts of the network configuration manually,
# please utilize the 'source' or 'source-directory' directives to do
# so.
# PVE will preserve these directives, but will NOT read its network
# configuration from sourced files, so do not attempt to move any of
# the PVE managed interfaces into external files!

auto lo
iface lo inet loopback

iface ens3 inet manual

auto vmbro
iface vmbro inet static
    address 192.168.0.1/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0
```

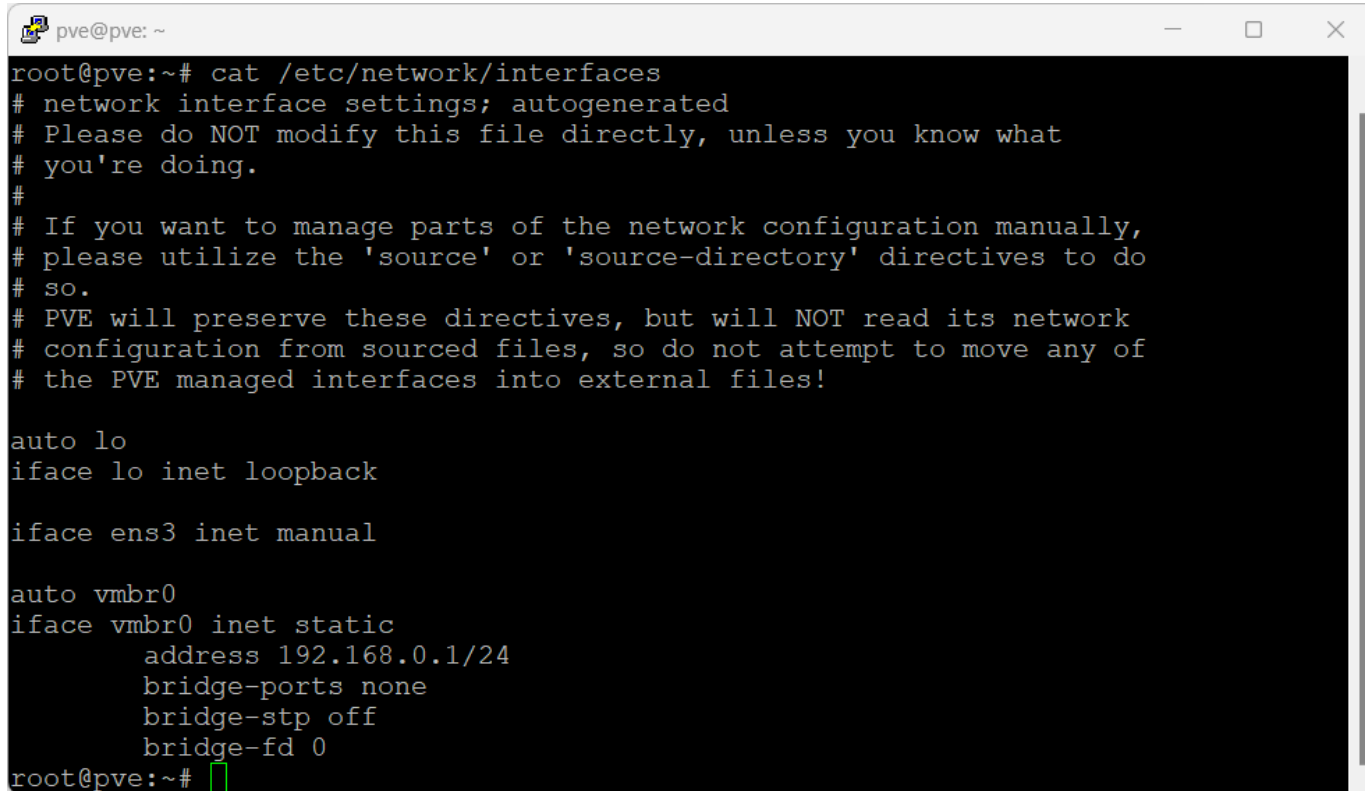
Simpan perubahan dengan Menekan tombol **CTRL+O** dan tekan **Enter**.

Tekan tombol **CTRL+X** untuk Keluar dari editor **nano**.

MENGATUR INTERFACE BRIDGE VMBRO PADA FILE /ETC/NETWORK/INTERFACES

Memverifikasi hasil penyesuaian pada *file* **/etc/network/interfaces**.

```
cat /etc/network/interfaces
```

A terminal window titled 'pve@pve: ~' with standard window controls. The terminal displays the output of the command 'cat /etc/network/interfaces'. The output shows a series of comments explaining the file's purpose and usage, followed by configuration for three interfaces: 'lo' (loopback), 'ens3' (manual), and 'vmbr0' (static IP 192.168.0.1/24 with bridge settings). The prompt 'root@pve:~#' is visible at the bottom with a green cursor.

```
root@pve:~# cat /etc/network/interfaces
# network interface settings; autogenerated
# Please do NOT modify this file directly, unless you know what
# you're doing.
#
# If you want to manage parts of the network configuration manually,
# please utilize the 'source' or 'source-directory' directives to do
# so.
# PVE will preserve these directives, but will NOT read its network
# configuration from sourced files, so do not attempt to move any of
# the PVE managed interfaces into external files!

auto lo
iface lo inet loopback

iface ens3 inet manual

auto vmbr0
iface vmbr0 inet static
    address 192.168.0.1/24
    bridge-ports none
    bridge-stp off
    bridge-fd 0
root@pve:~#
```

MENGATUR INTERFACE BRIDGE VMBRO PADA FILE /ETC/NETWORK/INTERFACES

- Melakukan **restart service networking** agar perubahan yang dilakukan pada *file* tersebut diaktifkan.

```
systemctl restart networking
```

- Menampilkan konfigurasi *interface* jaringan.

```
ip link
```

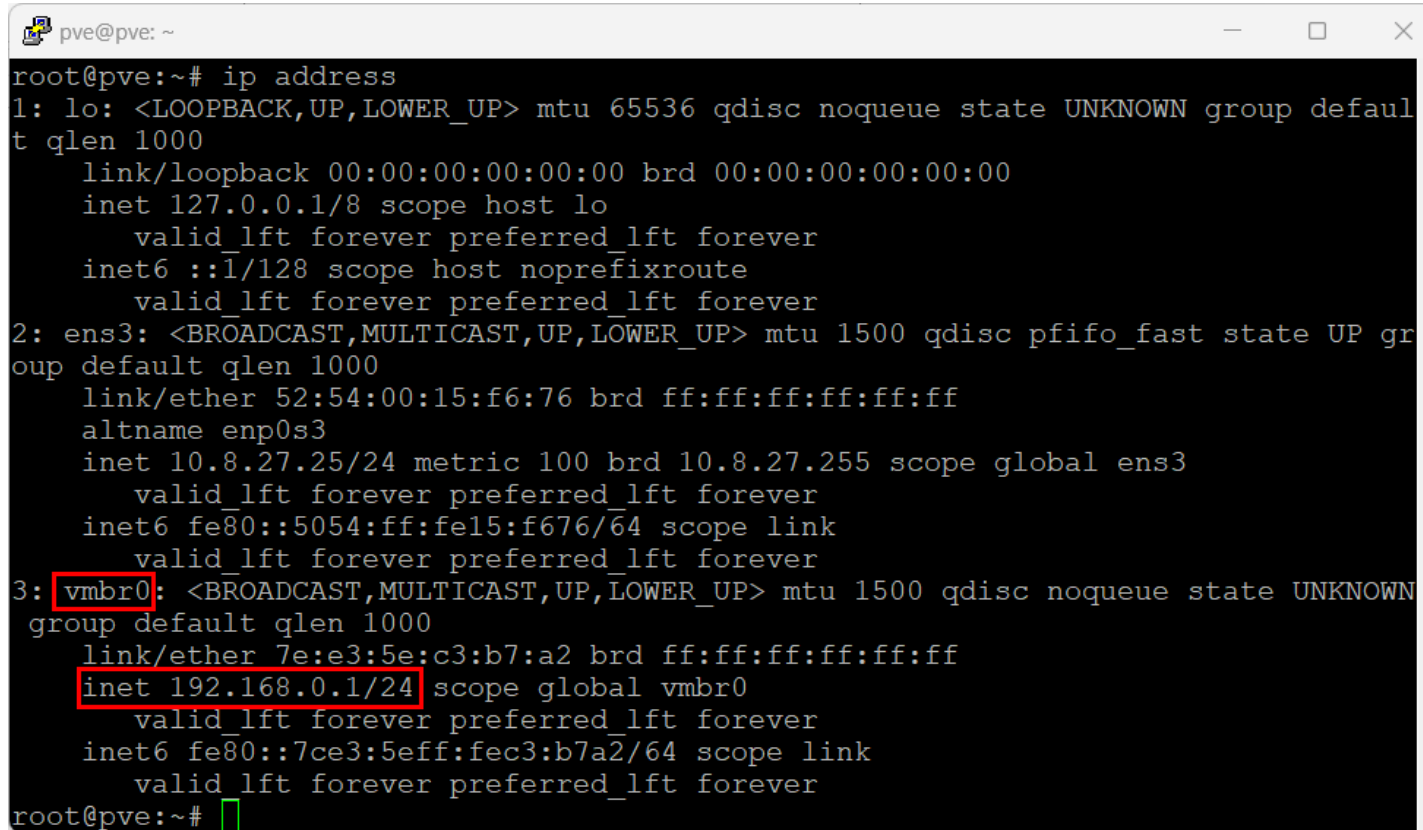
```
root@pve:~# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
   group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mo
   de DEFAULT group default qlen 1000
    link/ether 52:54:00:15:f6:76 brd ff:ff:ff:ff:ff:ff
    altnam enp0s3
3: vmbro: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
   mode DEFAULT group default qlen 1000
    link/ether 7e:e3:5e:c3:b7:a2 brd ff:ff:ff:ff:ff:ff
```

Terlihat *interface vmbro* berhasil dibuat.

MENGATUR INTERFACE BRIDGE VMBRO PADA FILE /ETC/NETWORK/INTERFACES

Menampilkan informasi pengalamatan IP pada setiap *interface* jaringan

ip address



```
pve@pve: ~  
root@pve:~# ip address  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000  
    link/ether 52:54:00:15:f6:76 brd ff:ff:ff:ff:ff:ff  
    altname enp0s3  
    inet 10.8.27.25/24 metric 100 brd 10.8.27.255 scope global ens3  
        valid_lft forever preferred_lft forever  
    inet6 fe80::5054:ff:fe15:f676/64 scope link  
        valid_lft forever preferred_lft forever  
3: vmbro: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default qlen 1000  
    link/ether 7e:e3:5e:c3:b7:a2 brd ff:ff:ff:ff:ff:ff  
    inet 192.168.0.1/24 scope global vmbro  
        valid_lft forever preferred_lft forever  
    inet6 fe80::7ce3:5eff:fec3:b7a2/64 scope link  
        valid_lft forever preferred_lft forever  
root@pve:~#
```

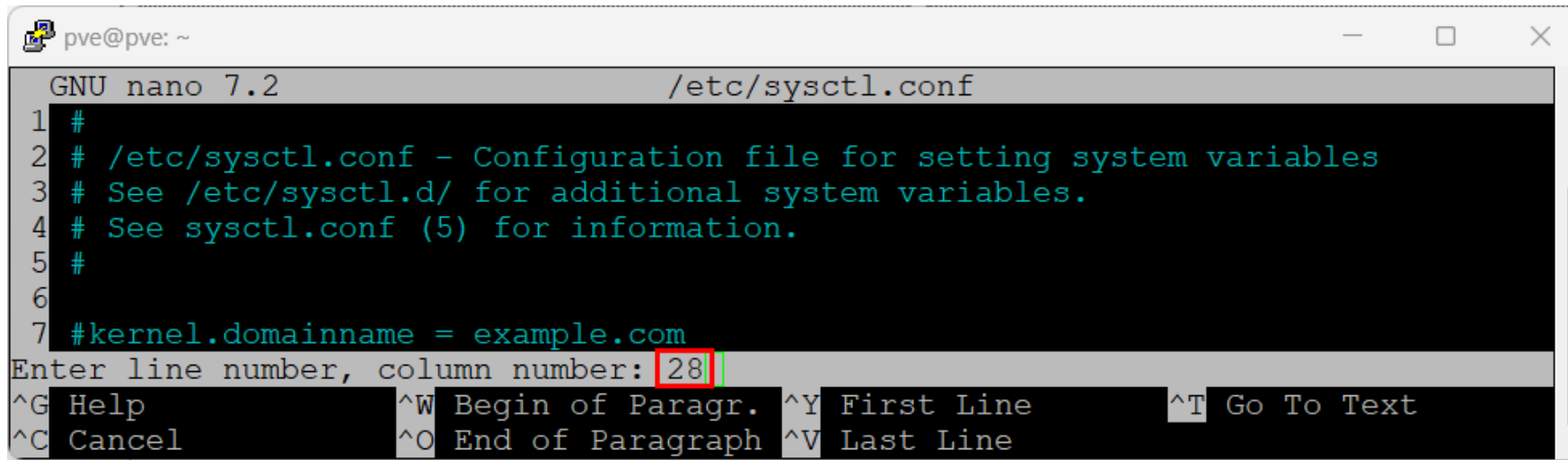
Terlihat *interface* **vmbro** menggunakan alamat IP **192.168.0.1/24**.

MENGAKTIFKAN FITUR IP FORWARDING PADA FILE /ETC/SYSCTL.CONF

Pengaktifan fitur **IP Forwarding** dilakukan agar PVE dapat merutekan paket data antar jaringan sehingga bertindak sebagai **router**.

```
nano -l /etc/sysctl.conf
```

Berpindah ke baris **28** yang memuat parameter **net.ipv4.ip_forward=1**.



```
pve@pve: ~  
GNU nano 7.2 /etc/sysctl.conf  
1 #  
2 # /etc/sysctl.conf - Configuration file for setting system variables  
3 # See /etc/sysctl.d/ for additional system variables.  
4 # See sysctl.conf (5) for information.  
5 #  
6  
7 #kernel.domainname = example.com  
Enter line number, column number: 28  
^G Help      ^W Begin of Paragr. ^Y First Line    ^T Go To Text  
^C Cancel    ^O End of Paragraph ^V Last Line
```

Tekan **CTRL+_** dan masukkan nomor baris **28** pada inputan **Enter line number, column number:** serta tekan **Enter**.

MENGAKTIFKAN FITUR IP FORWARDING PADA FILE /ETC/SYSCTL.CONF

Tampil baris 28 yang memuat nilai `#net.ipv4.ip_forward=1`

```
GNU nano 7.2 /etc/sysctl.conf
25 #net.ipv4.tcp_syncookies=1
26
27 # Uncomment the next line to enable packet forwarding for IPv4
28 #net.ipv4.ip_forward=1
29
30 # Uncomment the next line to enable packet forwarding for IPv6
31 # Enabling this option disables Stateless Address Autoconfiguration
```

Hilangkan tanda `#` di awal baris 28 tersebut untuk mengaktifkan fitur *IP forwarding* sehingga terlihat seperti berikut:

```
GNU nano 7.2 /etc/sysctl.conf *
25 #net.ipv4.tcp_syncookies=1
26
27 # Uncomment the next line to enable packet forwarding for IPv4
28 net.ipv4.ip_forward=1
29
30 # Uncomment the next line to enable packet forwarding for IPv6
31 # Enabling this option disables Stateless Address Autoconfiguration
```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MENGAKTIFKAN FITUR IP FORWARDING PADA FILE /ETC/SYSCTL.CONF

Mengaktifkan perubahan pada *file* **/etc/sysctl.conf**.

`sysctl -p`

```
root@pve:~# sysctl -p
net.ipv4.ip_forward = 1
```

MENGATUR INTERNET CONNECTION SHARING (ICS) PADA PVE MENGGUNAKAN IPTABLES

Internet Connection Sharing (ICS) atau berbagi pakai koneksi Internet dapat dilakukan pada PVE dengan mengatur **IPTables** tabel **NAT (Network Address Translation)** sehingga VM/CT dapat terkoneksi ke Internet.

```
iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

```
root@pve:~# iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

MENGATUR INTERNET CONNECTION SHARING (ICS) PADA PVE MENGGUNAKAN IPTABLES

Memverifikasi hasil pengaturan **IPTables**.

```
iptables -t nat -L --line-numbers
```

```
root@pve:~# iptables -t nat -L --line-numbers
Chain PREROUTING (policy ACCEPT)
num  target      prot opt source                destination

Chain INPUT (policy ACCEPT)
num  target      prot opt source                destination

Chain OUTPUT (policy ACCEPT)
num  target      prot opt source                destination

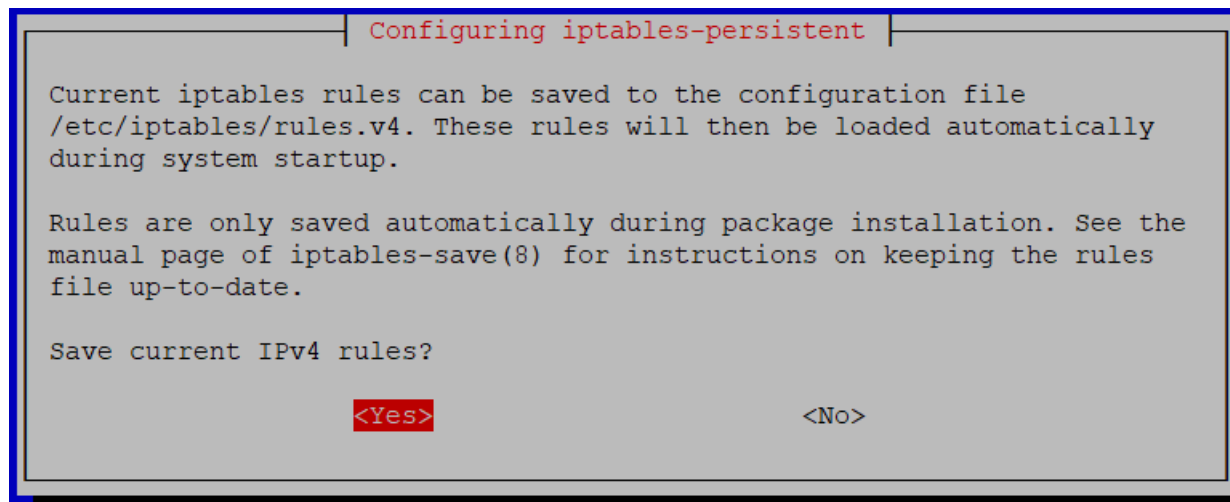
Chain POSTROUTING (policy ACCEPT)
num  target      prot opt source                destination
1    MASQUERADE  all  --  anywhere              anywhere
```

MENGINSTALASI PAKET IPTABLES-PERSISTENT

Instalasi paket **iptables-persistent** diperlukan agar konfigurasi **IPTables** tersimpan secara permanen dan ketika *booting (startup)* diaktifkan secara otomatis.

```
apt -y install iptables-persistent
```

Tampil kotak dialog **Configuring iptables-persistent** yang menampilkan pesan konfirmasi untuk menyimpan aturan **IPTables IPv4** yang aktif saat ini (**IPv4 rules**):



Tekan **Enter** untuk menyimpan aturan **IPTables IPv4** yang aktif saat ini. Aturan ini hanya disimpan secara otomatis pada saat instalasi paket **iptables-persistent**. Penyimpanan berikutnya dapat dilakukan dengan menggunakan perintah **iptables-save**.

MENGINSTALASI PAKET IPTABLES-PERSISTENT

Tampil pesan konfirmasi untuk menyimpan aturan **IPTables IPv6** yang aktif saat ini (**IPv6 rules**):

```
Configuring iptables-persistent

Current iptables rules can be saved to the configuration file
/etc/iptables/rules.v6. These rules will then be loaded automatically
during system startup.

Rules are only saved automatically during package installation. See the
manual page of ip6tables-save(8) for instructions on keeping the rules
file up-to-date.

Save current IPv6 rules?

<Yes> <No>
```

Tekan **tab** untuk berpindah ke tombol **<No>** dan tekan **Enter** agar tidak menyimpan aturan **IPTables IPv6**. Tunggu hingga proses instalasi selesai dilakukan.

MENGINSTALASI PAKET IPTABLES-PERSISTENT

Memverifikasi hasil penyimpanan konfigurasi **IPTables**.

```
cat /etc/iptables/rules.v4
```

```
root@pve:~# cat /etc/iptables/rules.v4
# Generated by iptables-save v1.8.9 on Mon May 12 10:31:37 2025
*filter
:INPUT ACCEPT [14014:144060632]
:FORWARD ACCEPT [32:13689]
:OUTPUT ACCEPT [12301:4072023]
COMMIT
# Completed on Mon May 12 10:31:37 2025
# Generated by iptables-save v1.8.9 on Mon May 12 10:31:37 2025
*raw
:PREROUTING ACCEPT [14049:144075501]
:OUTPUT ACCEPT [12301:4072023]
COMMIT
# Completed on Mon May 12 10:31:37 2025
# Generated by iptables-save v1.8.9 on Mon May 12 10:31:37 2025
*nat
:PREROUTING ACCEPT [1:224]
:INPUT ACCEPT [1:224]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens3 -j MASQUERADE
COMMIT
# Completed on Mon May 12 10:31:37 2025
```

MENONAKTIFKAN PESAN NOTIFIKASI NO VALID SUBSCRIPTION PADA PVE 8.4.1

Mengubah parameter pada file `"/usr/share/javascript/proxmox-widgettoolkit/proxmoxlib.js"` yaitu: **Ext.Msg.show({ title: gettext('No valid sub** menjadi **void({ //Ext.Msg.show({** pada baris 549 dari file tersebut.

Selain itu juga sekaligus membuat *file backup* dengan nama **“proxmoxlib.js.bak”** dan melakukan *restart service pveproxy.service* untuk mengaktifkan perubahan tersebut yaitu dengan mengeksekusi perintah berikut:

```
sed -Ezi.bak "s/(Ext.Msg.show\\(\\{\\s+title: gettext\\('No valid sub)/void\\(\\{ \\/\\/\\1/g" \  
/usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js \  
&& systemctl restart pveproxy.service
```

```
root@pve:~# sed -Ezi.bak "s/(Ext.Msg.show\\(\\{\\s+title: gettext\\('No valid sub)/void\\(\\{ \\/\\/\\1/g" \  
> /usr/share/javascript/proxmox-widget-toolkit/proxmoxlib.js \  
> && systemctl restart pveproxy.service
```

MANAJEMEN REPOSITORY PADA PVE 8.4.1

- Menurut *Wiki* dari *Proxmox*, **repository** merupakan kumpulan paket perangkat lunak yang dapat digunakan untuk menginstalasi perangkat lunak baru dan memperoleh pembaharuan terkait paket perangkat lunak yang telah terinstalasi.
- Secara *default* **PVE Enterprise Repository** telah aktif serta direkomendasikan untuk tetap digunakan bagi pengguna PVE yang melakukan *subscription* karena memuat paket yang paling stabil sehingga sangat cocok digunakan untuk *production*.
- Untuk dapat memanfaatkan *repository* ini maka diperlukan *subscription key* yang **BERBAYAR**. Detail informasi pembiayaan terkait *PVE subscription* dapat dilihat pada alamat <https://www.proxmox.com/en/proxmox-ve/pricing>.

MENONAKTIFKAN PVE ENTERPRISE SUBSCRIPTION

- Menonaktifkan **PVE Enterprise Repository** dengan cara menambahkan tanda **#** diawal baris dari konten pada file `/etc/apt/sources.list.d/pve-enterprise.list` menggunakan editor **nano**.

```
nano /etc/apt/sources.list.d/pve-enterprise.list
```

```
GNU nano 7.2 /etc/apt/sources.list.d/pve-enterprise.list *  
#deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise
```

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

- Memverifikasi hasil penonaktifan dengan mengeksekusi perintah:

```
# cat /etc/apt/sources.list.d/pve-enterprise.list
```

```
root@pve:~# cat /etc/apt/sources.list.d/pve-enterprise.list  
#deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise
```

Terlihat pada awal baris dari file **pve-enterprise.list** telah terdapat tanda **#** yang berfungsi sebagai komentar.

MENGUJI PENGATURAN REPOSITORY

Secara *default* telah ada pengaturan **PVE No-Subscription Repository** di **/etc/apt/sources.list.d/pve-install-repo.list**.

```
root@pve:~# cat /etc/apt/sources.list.d/pve-install-repo.list
deb [arch=amd64] http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

Untuk menguji hasil pengaturan *repository* yang telah dilakukan maka eksekusi perintah:

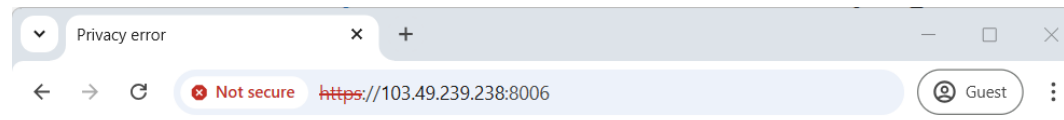
```
apt update
```

MENGAKSES WEB GUI DARI PVE

- Buka **browser**, sebagai contoh menggunakan **Chrome**. Pada **address bar** dari **browser**, masukkan **URL** <https://103.49.239.238:8006>

Ganti nilai 103.49.239.238 dengan alamat IP dari VPS yang dimiliki.

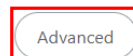
- Hasil pengaksesan akan menampilkan pesan peringatan “**Your connection is not private**”.



Your connection is not private

Attackers might be trying to steal your information from **103.49.239.238** (for example, passwords, messages, or credit cards). [Learn more about this warning](#)

NET::ERR_CERT_AUTHORITY_INVALID

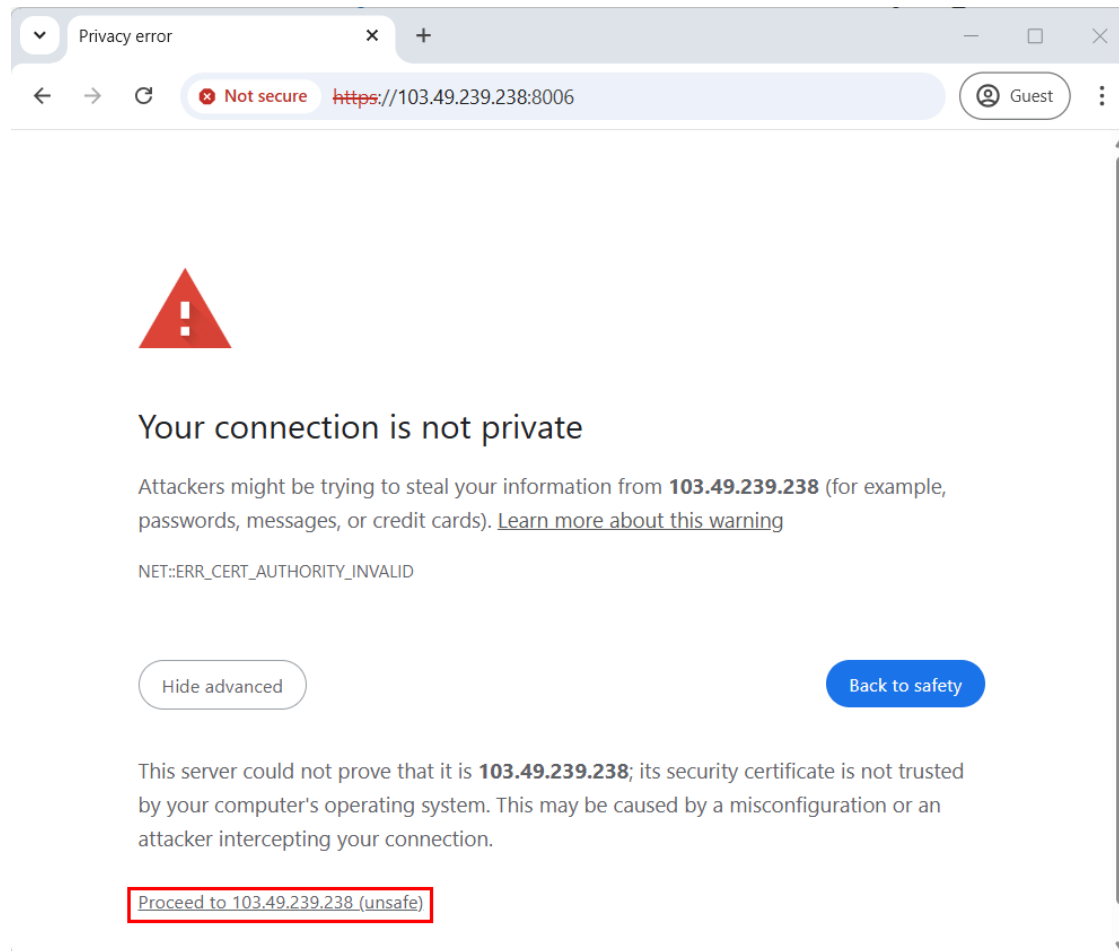


Back to safety

Klik tombol **Advanced** untuk melanjutkan pengaksesan.

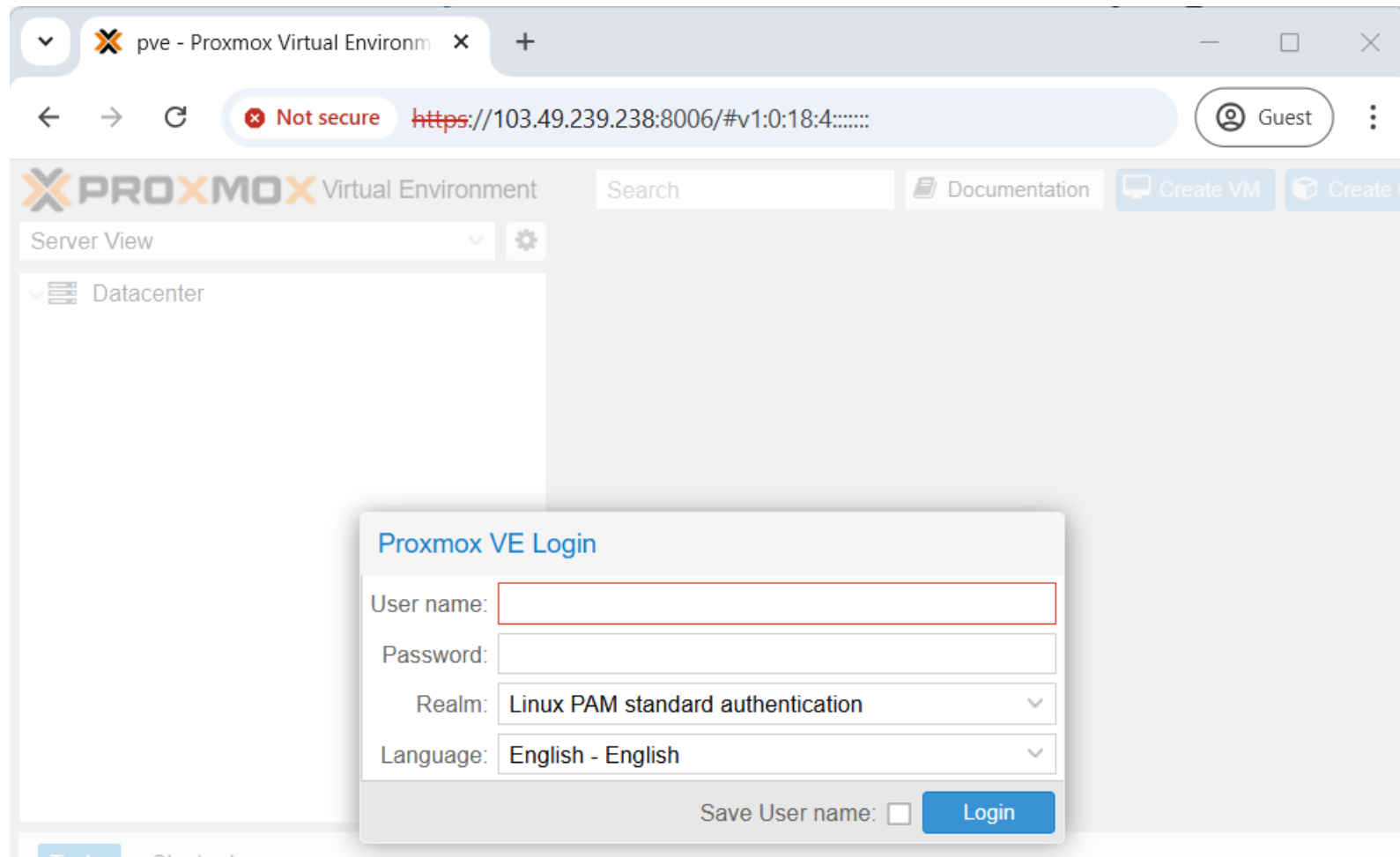
MENGAKSES WEB GUI DARI PVE

Klik *link* “**Proceed to 103.49.239.238 (unsafe)**” untuk melanjutkan proses.



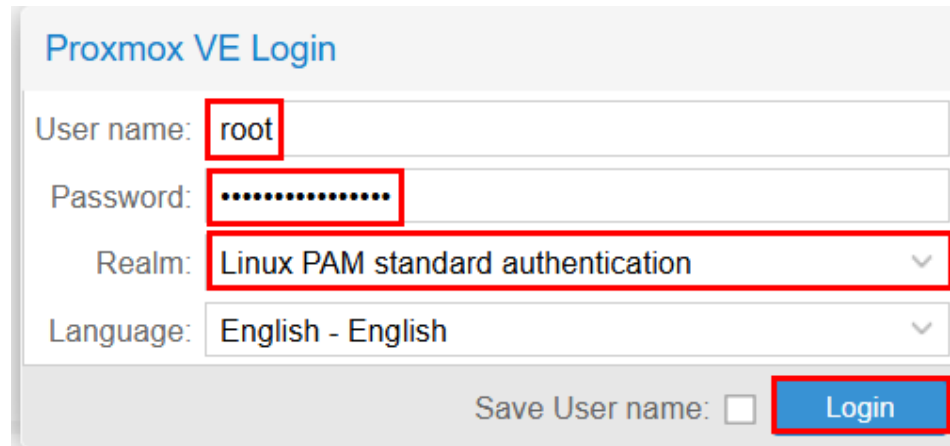
MENGAKSES WEB GUI DARI PVE

Tampil halaman **Login** dari Web GUI PVE.



MENGAKSES WEB GUI DARI PVE

- Lengkapi inputan **User name** dengan nilai “**root**” dan **Password** dari *user root* tersebut serta pastikan **Realm** menggunakan **Linux PAM standard authentication**.



The screenshot shows the 'Proxmox VE Login' form. The 'User name' field contains 'root', the 'Password' field is filled with dots, and the 'Realm' dropdown menu is set to 'Linux PAM standard authentication'. The 'Language' dropdown is set to 'English - English'. At the bottom, there is a 'Save User name' checkbox and a blue 'Login' button. Red rectangular boxes highlight the 'User name', 'Password', 'Realm', and 'Login' button.

Tekan tombol **Login** untuk memproses otentikasi.

MENGAKSES WEB GUI DARI PVE

Apabila proses otentikasi login berhasil dilakukan maka akan tampil halaman **Dashboard** dari **Web GUI PVE**

PROXMOX Virtual Environment 8.4.1

Search

Documentation Create VM Create CT root@pam

Server View Datacenter

Datacenter

pve

Search

Summary Notes Cluster Ceph

Type ↑	Description	Disk usage...	Memory us...	CPU usage
node	pve	5.9 %	36.8 %	0.9% of 2 ...
sdn	localnetwork (pve)			
storage	local (pve)	5.9 %		

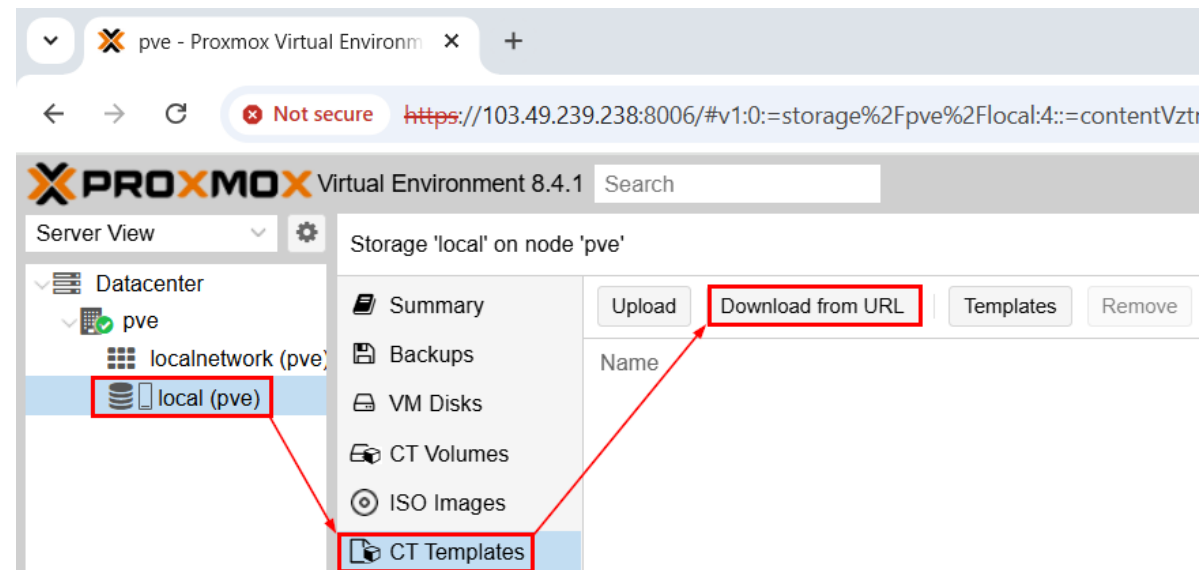
MANAJEMEN LINUX CONTAINER (LXC) DI PVE

PENGENALAN LINUX CONTAINER (LXC)

- Menurut situs [Proxmox](#), LXC merupakan lingkungan virtualisasi level sistem operasi untuk menjalankan beberapa sistem *Linux* terisolasi pada sebuah kontrol *host Linux*.
- LXC menjadi alternatif dari **full machine virtualization** yang menawarkan **low overhead**.
- *Container* akan menggunakan sistem operasi dari *host* daripada mengemulasikan sistem operasi secara lengkap sehingga berdampak pada keseluruhan container menggunakan *kernel* yang sama dan dapat mengakses sumber daya secara langsung dari *host*.
- Pengguna *Linux* dapat membuat dan manajemen *container* sistem atau aplikasi menggunakan **Application Programming Interface (API)**.

MENGUNDUH FILE CONTAINER (CT) TEMPLATE IMAGE UBUNTU KE LOCAL STORAGE DARI PVE

- CT yang akan dibuat menggunakan sistem operasi **Ubuntu 24.04** dengan **template image** yang diunduh dari situs **Proxmox** pada alamat <http://download.proxmox.com/images/system/>
- Panel sebelah kiri dari *Web GUI PVE*, pilih **storage local (pve)** yang terdapat dibawah **node pve** dari menu **Datacenter**. Selanjutnya pilih menu **CT Templates** pada panel sebelah kanan dari **local (pve)** dan tekan tombol **Download from URL**.

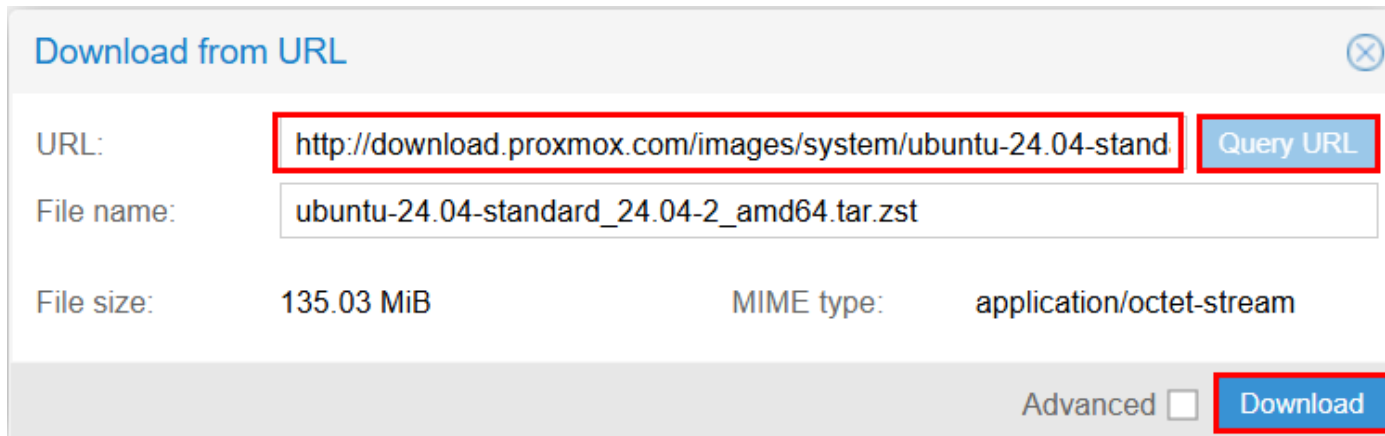


MENGUNDUH FILE CONTAINER (CT) TEMPLATE IMAGE UBUNTU KE LOCAL STORAGE DARI PVE

Tampil kotak dialog **Download from URL**. Pada inputan **URL** masukkan alamat berikut:

http://download.proxmox.com/images/system/ubuntu-24.04-standard_24.04-2_amd64.tar.zst

Kemudian tekan tombol **Query URL** sehingga inputan **File name:** akan otomatis terisi dengan nama *file* yaitu **ubuntu-24.04-standard_24.04-2_amd64.tar.zst**.



Download from URL

URL:

File name:

File size: 135.03 MiB MIME type: application/octet-stream

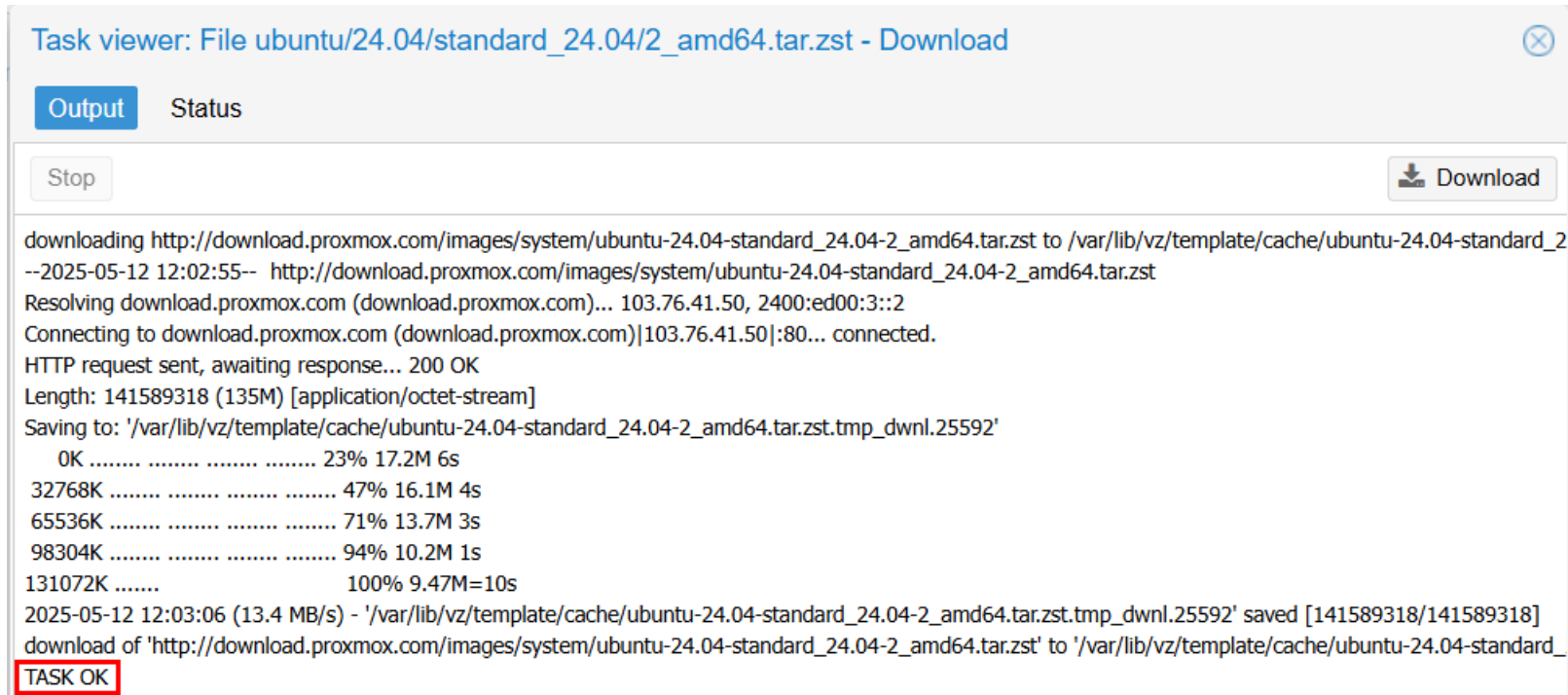
Advanced ☐

Tekan tombol **Download** untuk memproses pengunduhan.

MENGUNDUH FILE CONTAINER (CT) TEMPLATE IMAGE UBUNTU KE LOCAL STORAGE DARI PVE

Tampil kotak dialog **Task viewer: File**

ubuntu/24.04/standard_24.04/2_amd64.tar.zst - Download yang menampilkan *output* dari proses pengunduhan **file template image**. Tunggu hingga selesai yaitu ditandai dengan pesan **TASK OK**.



The screenshot shows a 'Task viewer' window titled 'Task viewer: File ubuntu/24.04/standard_24.04/2_amd64.tar.zst - Download'. It has tabs for 'Output' and 'Status', with 'Output' selected. A 'Stop' button is on the left and a 'Download' button is on the right. The main area displays the following log output:

```
downloading http://download.proxmox.com/images/system/ubuntu-24.04-standard_24.04-2_amd64.tar.zst to /var/lib/vz/template/cache/ubuntu-24.04-standard_2
--2025-05-12 12:02:55-- http://download.proxmox.com/images/system/ubuntu-24.04-standard_24.04-2_amd64.tar.zst
Resolving download.proxmox.com (download.proxmox.com)... 103.76.41.50, 2400:ed00:3::2
Connecting to download.proxmox.com (download.proxmox.com)[103.76.41.50]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 141589318 (135M) [application/octet-stream]
Saving to: '/var/lib/vz/template/cache/ubuntu-24.04-standard_24.04-2_amd64.tar.zst.tmp_dwnl.25592'
  OK ..... 23% 17.2M 6s
32768K ..... 47% 16.1M 4s
65536K ..... 71% 13.7M 3s
98304K ..... 94% 10.2M 1s
131072K ..... 100% 9.47M=10s
2025-05-12 12:03:06 (13.4 MB/s) - '/var/lib/vz/template/cache/ubuntu-24.04-standard_24.04-2_amd64.tar.zst.tmp_dwnl.25592' saved [141589318/141589318]
download of 'http://download.proxmox.com/images/system/ubuntu-24.04-standard_24.04-2_amd64.tar.zst' to '/var/lib/vz/template/cache/ubuntu-24.04-standard_
TASK OK
```

Tutup kotak dialog
Task viewer: File
ubuntu/24.04/stand
ard_24.04/2_amd64
.tar.zst

MENGUNDUH FILE CONTAINER (CT) TEMPLATE IMAGE UBUNTU KE LOCAL STORAGE DARI PVE

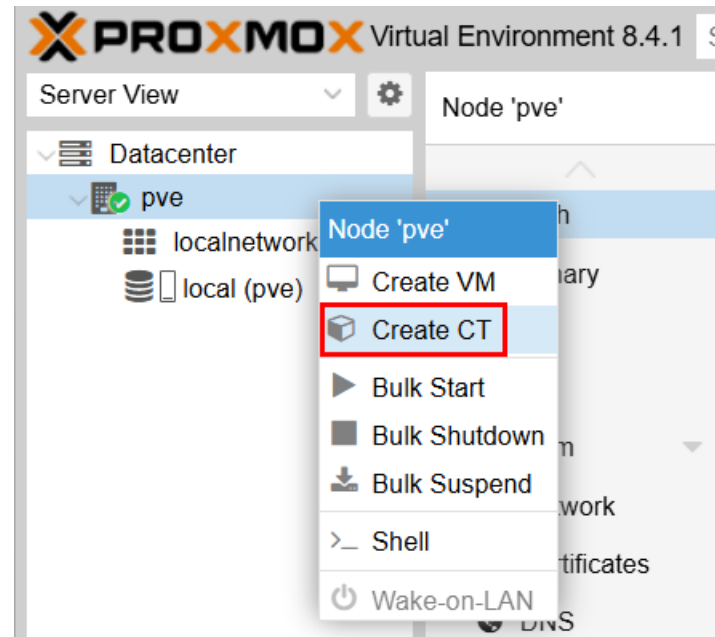
Hasil pengunduhan *file CT template image Ubuntu 24.04*.

The screenshot shows the Proxmox Virtual Environment (PVE) web interface. The browser address bar displays the URL: `https://103.49.239.238:8006/#v1:0:=storage%2Fpve%2Flocal:4::=contentVztmpl:....`. The interface title is "PROXMOX Virtual Environment 8.4.1". The left sidebar shows the "Server View" with a tree structure: "Datacenter" > "pve" > "localnetwork (pve)" > "local (pve)". The main content area is titled "Storage 'local' on node 'pve'" and shows a list of CT templates. The "CT Templates" tab is selected in the left sidebar. The table below shows the details of the downloaded template, which is highlighted with a red border.

Name	Date	Format	Size
ubuntu-24.04-standard_24.04-2_amd64.tar.zst	2025-05-12 12:03:06	tzst	141.59 MB

MEMBUAT CONTAINER (CT) DI PVE

Klik kanan pada *node pve* yang terdapat dibawah menu **Datacenter** dari panel sebelah kiri *Web GUI PVE* dan pilih **Create CT**.



MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog **Create: LXC Container**. Terdapat beberapa parameter yang diatur di bagian tab **General** yaitu:

The screenshot shows the 'Create: LXC Container' dialog box with the following fields and values:

- Node:** pve
- CT ID:** 100
- Hostname:** ansible.belajar.local
- Unprivileged container:** ☒
- Nesting:** ☒
- Resource Pool:** (empty)
- Password:** (masked with dots)
- Confirm password:** (masked with dots)
- SSH public key(s):** (empty)

At the bottom right, the **Next** button is highlighted with a red box.

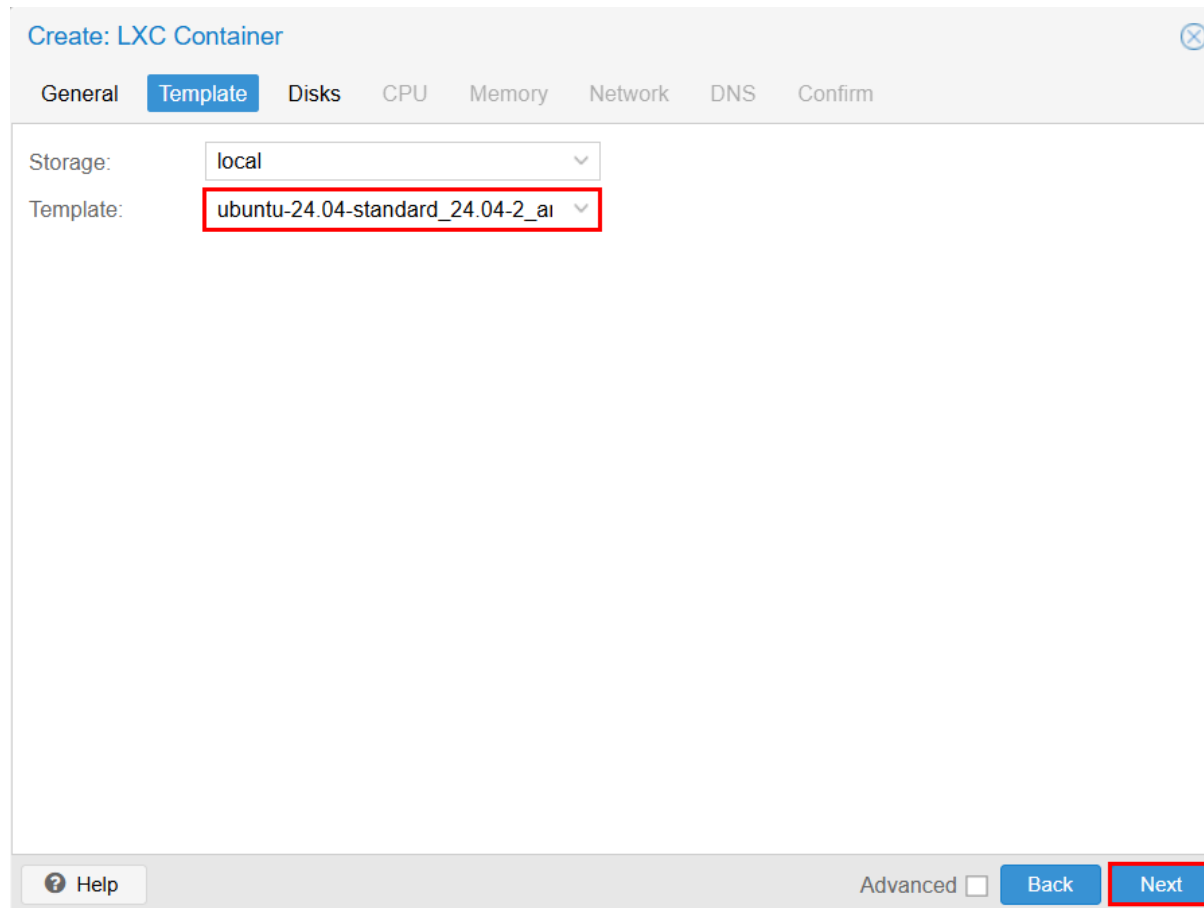
Lengkapi inputan **Hostname:** dengan nama pengenalan bagi CT yang dibuat, sebagai contoh **ansible.belajar.local**.

Sedangkan pada inputan **Password:** dan **Confirm password:**, masukkan sandi login dari user “**root**” untuk container *Ubuntu 24.04*, sebagai contoh “**mandalika**”.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog pengaturan bagian **Template** dari **LXC Container**. Pada *dropdown Template*, pilih **ubuntu-24.04-standard_24.04-2_amd64.tar.zst**.



Create: LXC Container

General **Template** Disks CPU Memory Network DNS Confirm

Storage: local

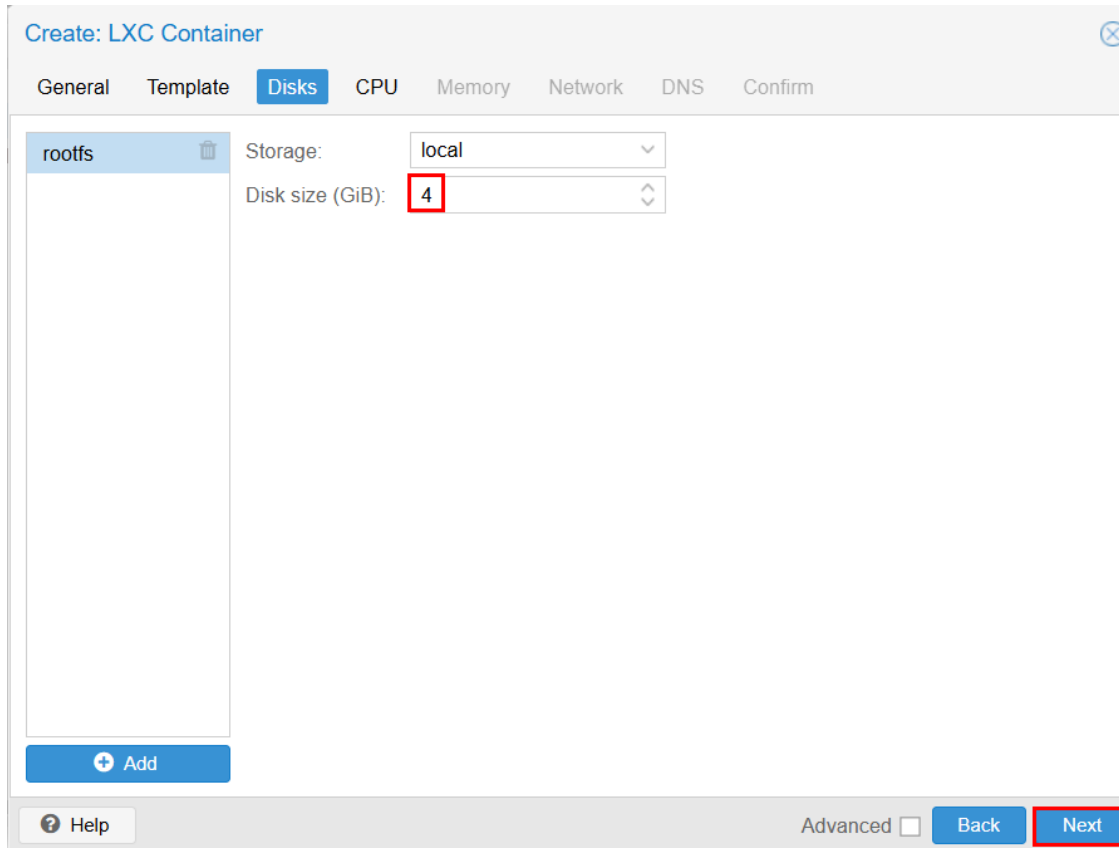
Template: **ubuntu-24.04-standard_24.04-2_ai**

Help Advanced ☐ Back **Next**

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

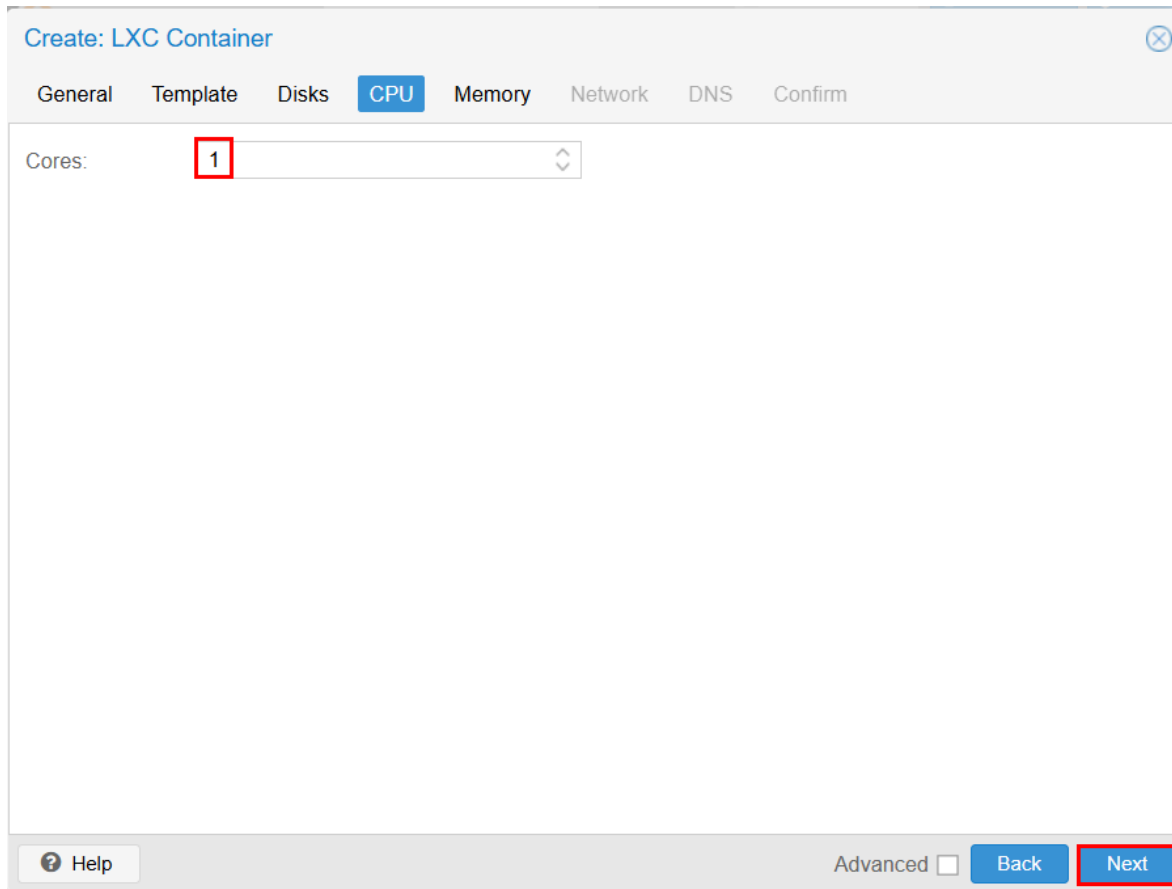
Tampil kotak dialog pengaturan bagian **Disks** dari **LXC Container**. Lakukan penyesuaian ukuran *hardisk* yang digunakan pada parameter **Disk size (GB)**: jika diperlukan. Secara *default* menggunakan **8 GB**. Sebagai contoh diubah menjadi **4 GB**.

The screenshot shows the 'Create: LXC Container' dialog box with the 'Disks' tab selected. On the left, there is a list of disks with one entry 'rootfs'. To the right of this list, the 'Storage' is set to 'local' and the 'Disk size (GiB)' is set to '4'. The '4' is highlighted with a red box. At the bottom left of the disk list is a blue '+ Add' button. At the bottom right of the dialog are 'Back' and 'Next' buttons, with the 'Next' button highlighted by a red box. Other tabs visible include General, Template, CPU, Memory, Network, DNS, and Confirm.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog pengaturan **CPU** dari **LXC Container**. Pada parameter **Cores:**, lakukan penyesuaian jumlah Core CPU yang digunakan apabila diperlukan. Secara *default* bernilai **1** (satu).

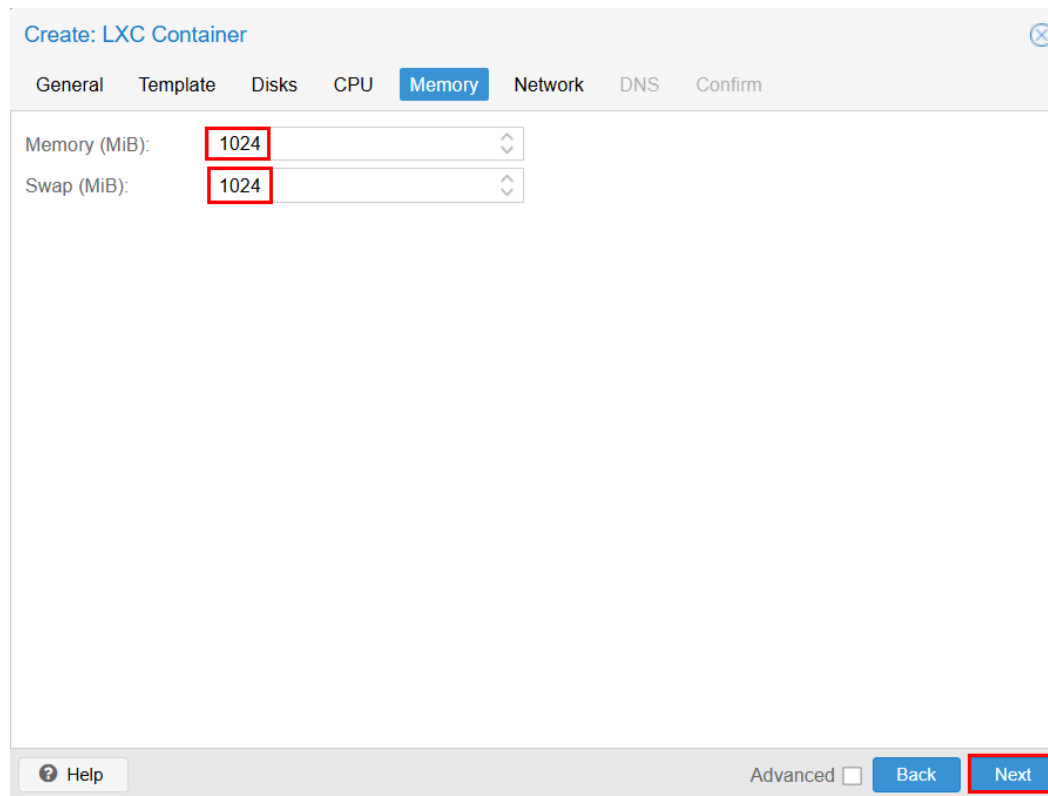


The screenshot shows the 'Create: LXC Container' dialog box with the 'CPU' tab selected. The 'Cores' dropdown menu is set to '1'. The 'Next' button is highlighted with a red box.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog pengaturan **Memory** dari **LXC Container**. Terdapat 2 (dua) parameter yang dapat diatur yaitu **Memory (MiB)** dan **Swap (MiB)**. Secara *default* masing-masing parameter tersebut bernilai **512 MiB**. Sebagai contoh dilakukan penyesuaian nilainya menjadi **1024 MiB** baik untuk parameter **Memory (MiB)** dan **Swap (MiB)**.

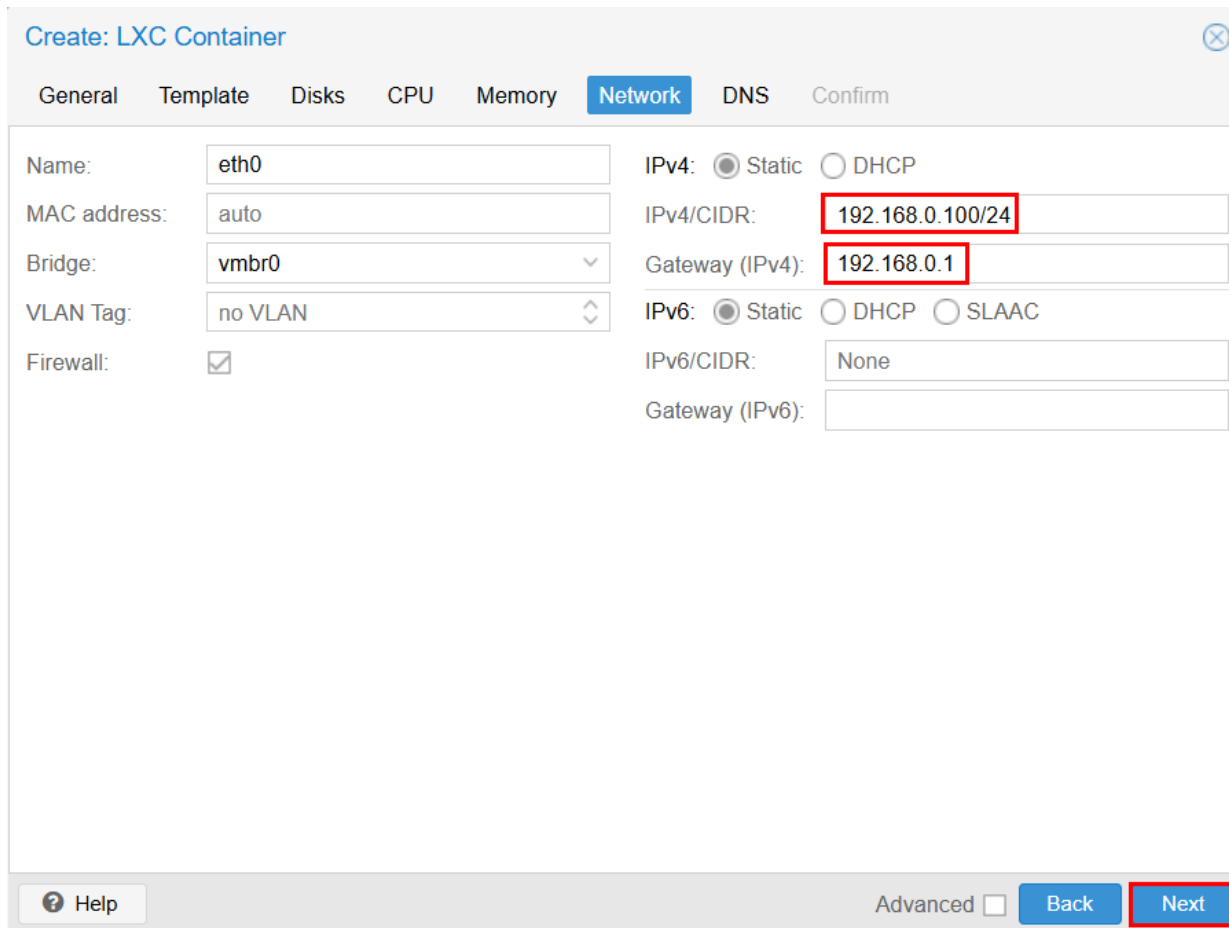


The screenshot shows the 'Create: LXC Container' dialog box with the 'Memory' tab selected. The 'Memory (MiB)' field is set to 1024, and the 'Swap (MiB)' field is also set to 1024. Both values are highlighted with red boxes. At the bottom right, the 'Next' button is also highlighted with a red box. The dialog includes tabs for General, Template, Disks, CPU, Memory, Network, DNS, and Confirm. A 'Help' button is at the bottom left, and an 'Advanced' checkbox is at the bottom center.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog pengaturan **Network** dari **LXC Container**. Secara default alokasi pengaturan pengalamatan **IPv4** pada *LXC Container* adalah secara **Static**.



The screenshot shows the 'Create: LXC Container' dialog box with the 'Network' tab selected. The configuration is as follows:

Field	Value
Name:	eth0
MAC address:	auto
Bridge:	vbr0
VLAN Tag:	no VLAN
Firewall:	☒
IPv4:	☒ Static ☐ DHCP
IPv4/CIDR:	192.168.0.100/24
Gateway (IPv4):	192.168.0.1
IPv6:	☒ Static ☐ DHCP ☐ SLAAC
IPv6/CIDR:	None
Gateway (IPv6):	

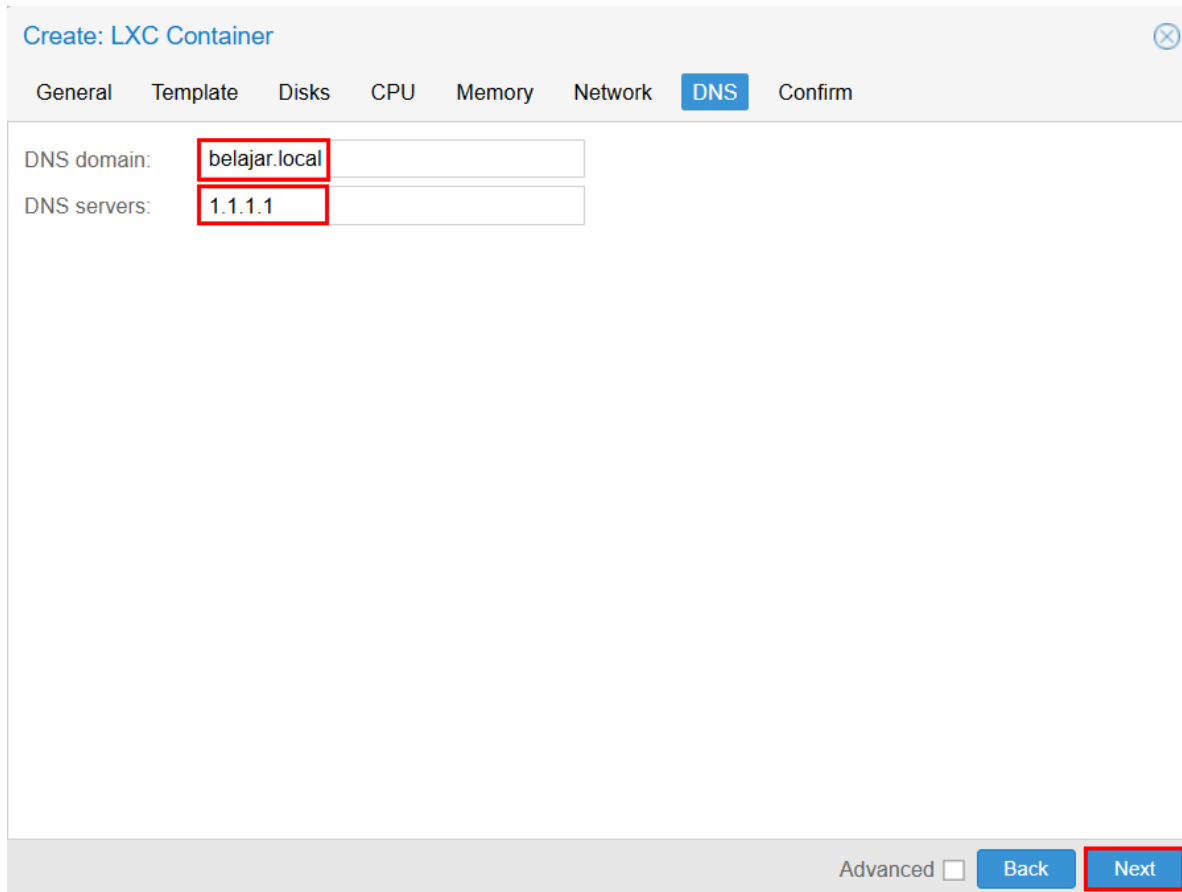
At the bottom of the dialog, there is a 'Next' button highlighted with a red box, along with a 'Back' button and an 'Advanced' checkbox.

Pada parameter **IPv4/CIDR**:
masukkan alamat IP dan subnetmask
yang digunakan oleh *container* yaitu
192.168.0.100/24. Sedangkan pada
bagian **Gateway (IPv4)**:, masukkan
alamat IP **192.168.0.1**.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog pengaturan **DNS** dari **LXC Container**. Pada inputan **parameter DNS domain**, lengkapi dengan nama domain yang digunakan. Sebagai contoh **belajar.local**.



The screenshot shows the 'Create: LXC Container' dialog box with the 'DNS' tab selected. The 'DNS domain' field contains 'belajar.local' and the 'DNS servers' field contains '1.1.1.1'. Both fields are highlighted with red boxes. The 'Next' button at the bottom right is also highlighted with a red box.

Sedangkan pada inputan parameter **DNS Servers**, lengkapi dengan alamat dari *Server DNS* yang digunakan. Sebagai contoh **1.1.1.1**.

Klik tombol **Next** untuk melanjutkan.

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog **Confirm** dari **LXC Container** yang menampilkan ringkasan pengaturan yang telah dilakukan terkait pembuatan CT. Klik tombol **Finish**.

Create: LXC Container

GeneralTemplateDisksCPUMemoryNetworkDNSConfirm

Key ↑	Value
cores	1
features	nesting=1
hostname	ansible.belajar.local
memory	1024
nameserver	1.1.1.1
net0	name=eth0,bridge=vbr0,firewall=1,ip=192.168.0.100/24,gw=192.168.0.1
nodename	pve
ostemplate	local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst
pool	
rootfs	local:4
searchdomain	belajar.local
ssh-public-keys	
swap	1024
unprivileged	1

☐ Start after created

Advanced ☐BackFinish

MEMBUAT CONTAINER (CT) DI PVE

Tampil kotak dialog **Task viewer: CT 100 – Create**. Tunggu hingga proses pembuatan *container* **Ubuntu 24.04** selesai dibuat yang ditandai dengan pesan “**TASK OK**” pada bagian **Output** dari kotak dialog **Task viewer: CT 100 – Create**.

Task viewer: CT 100 - Create

Output Status

Stop Download

```
Formatting '/var/lib/vz/images/100/vm-100-disk-0.raw', fmt=raw size=4294967296 preallocation=off
Creating filesystem with 1048576 4k blocks and 262144 inodes
Filesystem UUID: 8df63d21-03f6-4dac-b0c4-cadeea0b2b27
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736
extracting archive '/var/lib/vz/template/cache/ubuntu-24.04-standard_24.04-2_amd64.tar.zst'
Total bytes read: 564490240 (539MiB, 142MiB/s)
Detected container architecture: amd64
Creating SSH host key 'ssh_host_ed25519_key' - this may take some time ...
done: SHA256:WSKFjl5+4U5hTHFbbDwvGGxxOEUCxPilZwh3nUd/m1M root@ansible
Creating SSH host key 'ssh_host_ecdsa_key' - this may take some time ...
done: SHA256:P09I46X3iEOKM/SNEZEsqdSoFcZ6V5MzMt1cEAs29Go root@ansible
Creating SSH host key 'ssh_host_dsa_key' - this may take some time ...
done: SHA256:BF4QAU3XHIOjYCqXmVnJZxrZkQ9Si6rE3WJectsFAC root@ansible
Creating SSH host key 'ssh_host_rsa_key' - this may take some time ...
done: SHA256:B60630ZcGltl643BIFNe4ya8UirbMZ08hscK39I6Hs0 root@ansible
TASK OK
```

Tutup kotak dialog **Task viewer: CT 100 – Create**.

HASIL DARI PEMBUATAN CONTAINER (CT) DI PVE

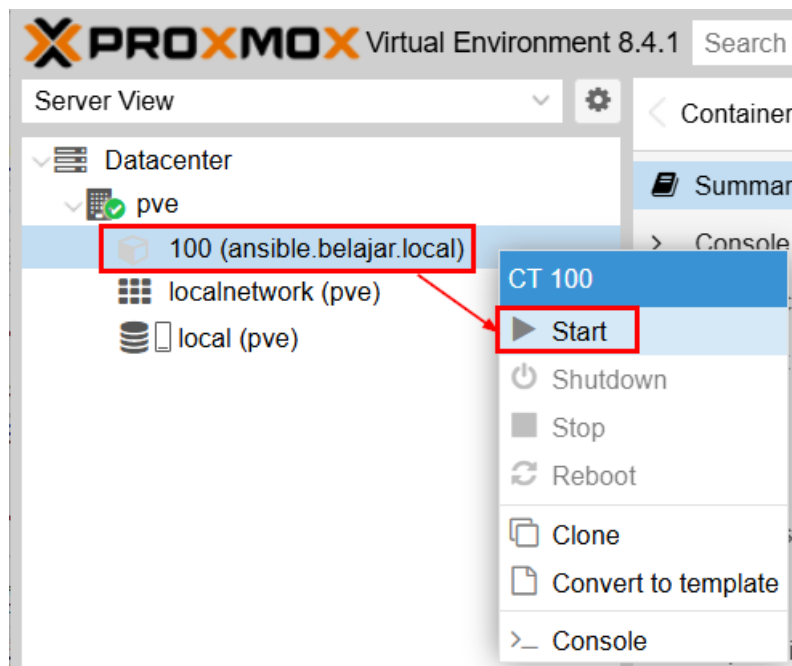
Terlihat **CT** dengan **ID 100** dan memiliki nama **ansible.belajar.local**.

The screenshot displays the Proxmox Virtual Environment (PVE) 8.4.1 web interface. The left sidebar shows the 'Server View' with a tree structure: 'Datacenter' > 'pve' > '100 (ansible.belajar.local)'. The container '100 (ansible.belajar.local)' is highlighted with a red rectangle. The main panel shows the 'Summary' tab for this container, which is an Ubuntu container on node 'pve'. The container is currently 'stopped'. The summary table lists various metrics:

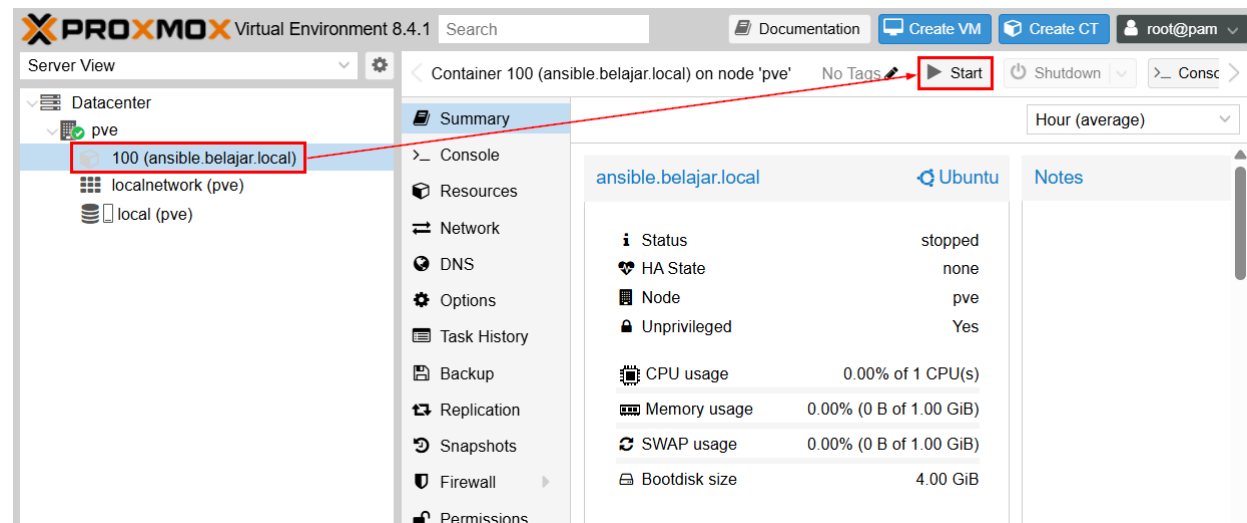
ansible.belajar.local	
Status	stopped
HA State	none
Node	pve
Unprivileged	Yes
CPU usage	0.00% of 1 CPU(s)
Memory usage	0.00% (0 B of 1.00 GiB)
SWAP usage	0.00% (0 B of 1.00 GiB)
Bootdisk size	4.00 GiB

MENJALANKAN CT ANSIBLE PADA PVE

Pada panel sebelah kiri dari *Web GUI PVE*, pilih **CT “100 (ansible.belajar.local)”** yang terdapat dibawah *node pve* dari menu **Datacenter**. Selanjutnya lakukan klik kanan pada **“100 (ansible.belajar.local)”** dan pilih **Start**.



Atau dengan cara memilih **“100 (ansible.belajar.local)”** yang terdapat dibawah *node pve* dari menu **Datacenter** dan pada panel detail sebelah kanan menekan tombol **Start**.



MENJALANKAN CT ANSIBLE PADA PVE

Apabila **CT “100 (ansible.belajar.local)”** telah berhasil dijalankan maka pada bagian **Tasks** dari **Log Panel** yang terdapat dibagian bawah dari **Web GUI PVE** akan menampilkan pesan **Status OK** untuk **CT 100 – Start**.

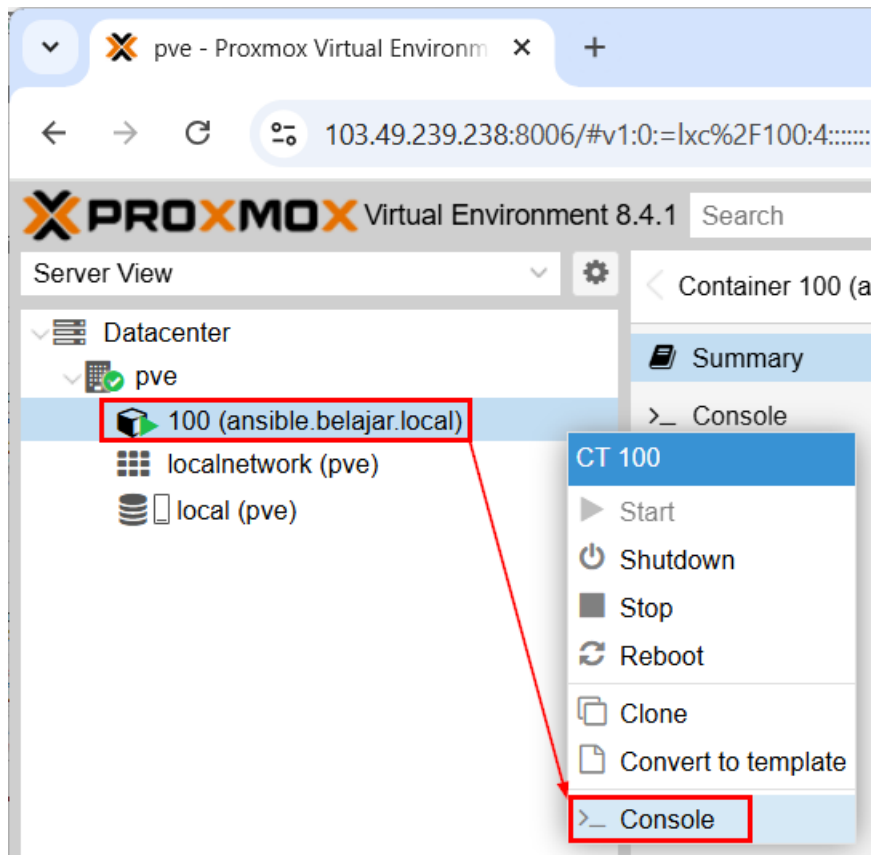
Tasks Cluster log						
Start Time ↓	End Time	Node	User name	Description	Status	
May 12 15:41:59	May 12 15:42:03	pve	root@pam	CT 100 - Start	OK	> ▲

Selain itu pada **CT “100 (ansible.belajar.local)”** yang terdapat dibawah *node pve* akan menampilkan simbol segitiga berwarna **hijau** sebagai penanda CT tersebut telah berjalan (**running**). Sedangkan pada panel detail sebelah kanan dari CT tersebut akan menampilkan **Status running**.

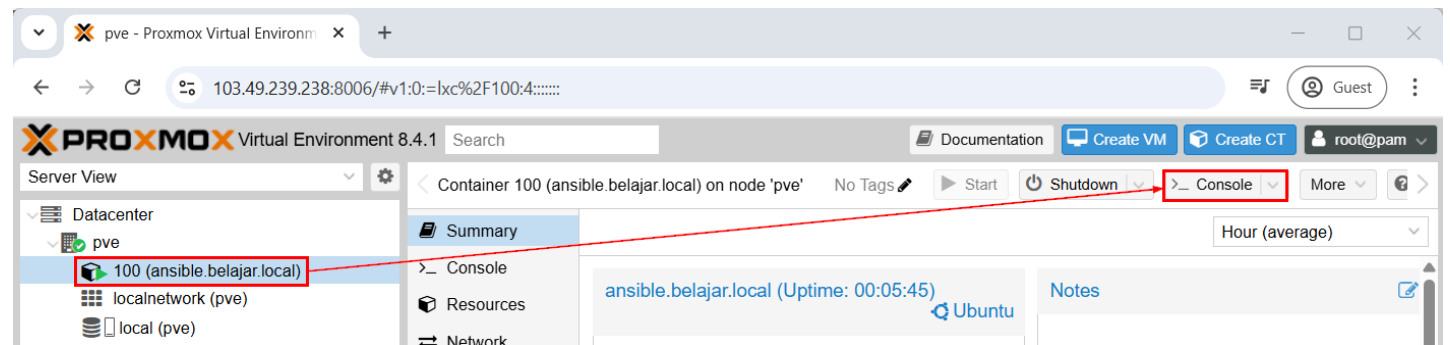
The screenshot shows the Proxmox Virtual Environment 8.4.1 interface. On the left, the 'Server View' sidebar shows the 'Datacenter' tree with 'pve' expanded, and '100 (ansible.belajar.local)' highlighted with a green triangle icon. The main panel displays the details for 'Container 100 (ansible.belajar.local) on node 'pve''. The 'Summary' tab is active, showing 'ansible.belajar.local (Uptime: 00:00:49)' with an 'Ubuntu' logo. A red box highlights the 'Status' field, which shows 'running'. Below it, the 'HA State' is shown as 'none'.

MENGAKSES CONSOLE DARI CT ANSIBLE PADA PVE

Klik kanan pada “100 (ansible.belajar.local)” di bawah *node pve* dari menu **Datacenter** dan pilih **Console**.

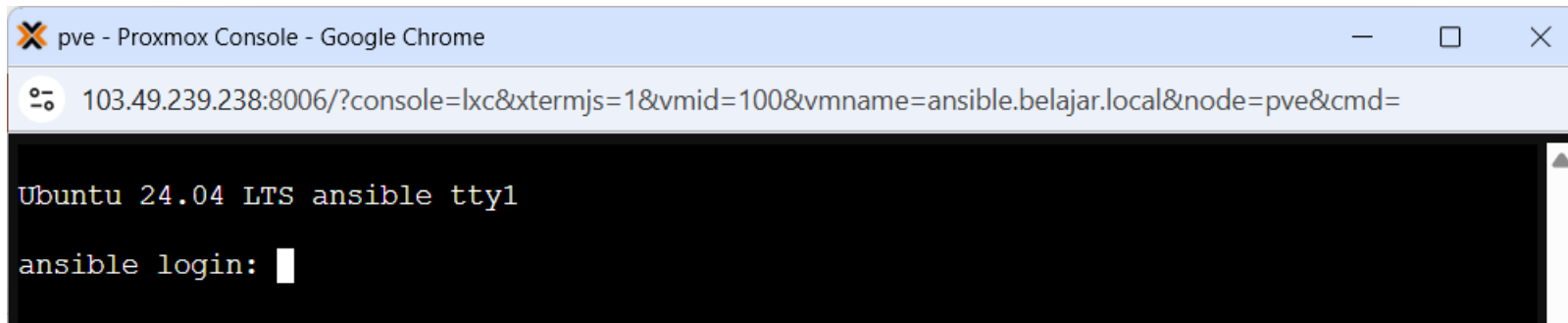


Atau dengan cara memilih “100 (ansible.belajar.local)” yang terdapat dibawah *node pve* dari menu **Datacenter** dan pada panel detail sebelah kanan menekan tombol **Console**.



KONFIGURASI CT ANSIBLE PADA PVE

Tampil inputan **ansible login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari CT **Ubuntu 24.04**.

A screenshot of a web browser window titled 'pve - Proxmox Console - Google Chrome'. The address bar shows the URL '103.49.239.238:8006/?console=lxc&xtermjs=1&vmid=100&vmname=ansible.belajar.local&node=pve&cmd='. The main content area is a black terminal window with white text. It displays 'Ubuntu 24.04 LTS ansible tty1' followed by the prompt 'ansible login:' and a white cursor.

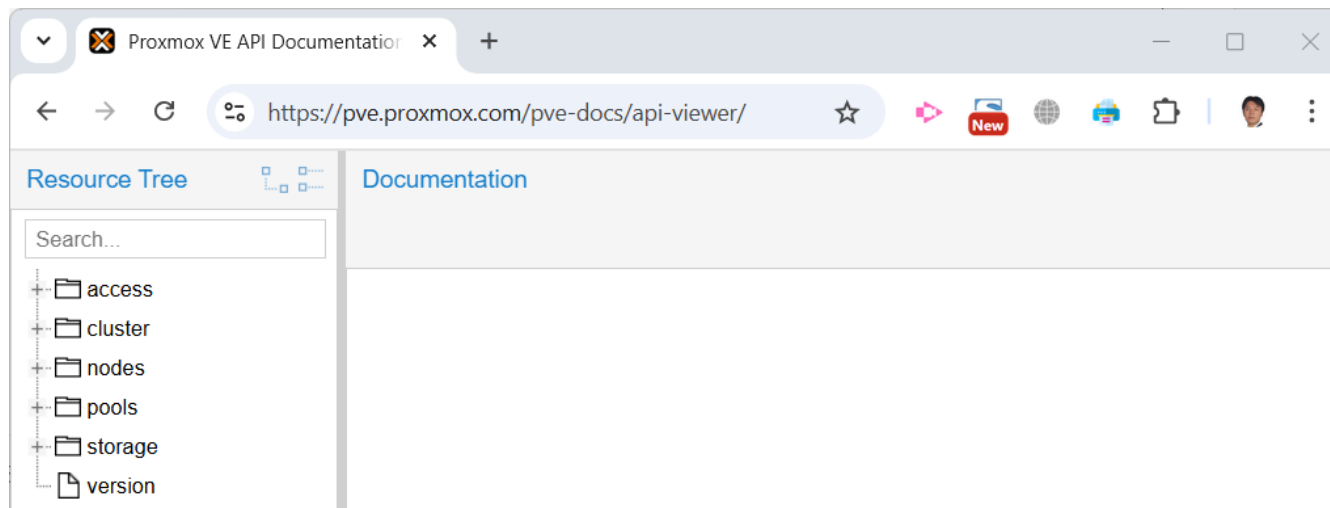
Masukkan nama login “**root**” pada inputan **ansible Login** dan tekan tombol **Enter**. Tampil inputan **Password:**, masukkan sandi login dari user “**root**” yaitu “**mandalika**”, dan tekan tombol **Enter**. Apabila proses otentikasi login berhasil dilakukan maka akan tampil **prompt CLI** dari CT **Ubuntu 24.04** yang ditandai dengan tanda **#**.

```
ansible login: root
Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.12-10-pve x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/pro
root@ansible:~#
```

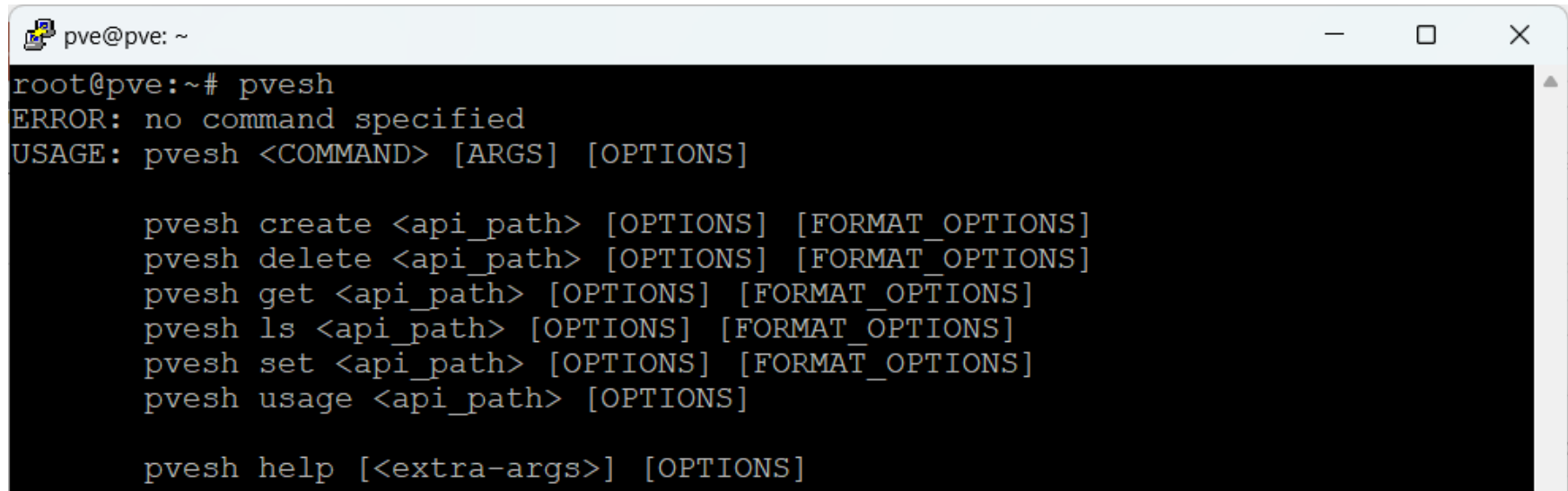
PVE APPLICATION PROGRAMMING INTERFACE (API)

- Menurut [wiki Proxmox](https://wiki.proxmox.com/), PVE menggunakan **Representational State Transfer (REST) Application Programming Interface (API)** dengan format data **JavaScript Object Notation (JSON)** yang dapat digunakan untuk mengontrol server PVE baik melalui **command line** maupun **Web Graphical User Interface (GUI)**.
- Secara formal API PVE didefinisikan menggunakan **JSON schema**.
- Dokumentasi API dari PVE yang memuat parameter dan *return value* dapat diakses pada alamat <https://pve.proxmox.com/pve-docs/api-viewer/>



PVE APPLICATION PROGRAMMING INTERFACE (API)

- API menggunakan **protokol HTTPS** dan **port 8006** dengan alamat akses yaitu **`https://ip.dari.server.pve:8006/api2/json/`**
- PVE menyediakan utilitas *command line* **pvesh** yang dapat digunakan untuk mengakses **REST API**. Hasil eksekusi dari perintah **pvesh** pada **terminal** dari **PVE** adalah sebagai berikut:

A terminal window titled 'pve@pve: ~' with standard window controls. The prompt is 'root@pve:~#'. The user has entered 'pvesh', which has resulted in an error message: 'ERROR: no command specified'. Below this is the usage information for the 'pvesh' command, showing various subcommands like 'create', 'delete', 'get', 'ls', 'set', 'usage', and 'help' with their respective arguments and options.

```
pve@pve: ~  
root@pve:~# pvesh  
ERROR: no command specified  
USAGE: pvesh <COMMAND> [ARGS] [OPTIONS]  
  
pvesh create <api_path> [OPTIONS] [FORMAT_OPTIONS]  
pvesh delete <api_path> [OPTIONS] [FORMAT_OPTIONS]  
pvesh get <api_path> [OPTIONS] [FORMAT_OPTIONS]  
pvesh ls <api_path> [OPTIONS] [FORMAT_OPTIONS]  
pvesh set <api_path> [OPTIONS] [FORMAT_OPTIONS]  
pvesh usage <api_path> [OPTIONS]  
  
pvesh help [<extra-args>] [OPTIONS]
```

PVE APPLICATION PROGRAMMING INTERFACE (API)

Terlihat bahwa utilitas **pvesh** memiliki 7 (tujuh) **command** yaitu **create**, **delete**, **get**, **ls**, **set** dan **usage** serta **help**.

- *Command* **create** digunakan untuk memanggil **API POST** sehingga dapat membuat sumber daya baru atau mengatur nilai pada objek yang belum diatur sebelumnya pada server PVE.
- *Command* **delete** digunakan untuk memanggil **API DELETE** sehingga dapat menghapus sumber daya tertentu pada server PVE.
- *Command* **get** digunakan untuk memanggil **API GET** sehingga dapat membaca atau mengambil informasi terkait objek pada server PVE.
- *Command* **ls** digunakan untuk menampilkan daftar *child object* pada **API PATH** tertentu.
- *Command* **set** digunakan untuk memanggil **API PUT** sehingga dapat memperbaharui nilai dari sumber daya atau objek pada server PVE.
- *Command* **usage** digunakan untuk menampilkan informasi penggunaan API dari **API PATH** tertentu.
- *Command* **help** digunakan untuk menampilkan informasi terkait tata cara penggunaan perintah tertentu dan argumen serta *options* yang dimiliki oleh perintah tersebut.

CONTOH PENGGUNAAN PVE APPLICATION PROGRAMMING INTERFACE (API)

Menampilkan informasi *storage* atau media penyimpanan yang dimiliki oleh server PVE menggunakan sintak penulisan:

```
pvesh get /nodes/{node}/storage
```

Ubah **{node}** dengan **hostname** dari server *proxmox* yaitu **pve** sehingga menjadi:

```
pvesh get /nodes/pve/storage
```

```
root@pve:~# pvesh get /nodes/pve/storage
```

content	storage	type	active	avail	enabled	shared	total	used	used_fraction
iso,backup,images,snippets,vztmpl,rootdir	local	dir	1	47.97 GiB	1	0	58.89 GiB	8.49 GiB	14.42%

Terlihat bahwa server **PVE** memiliki satu *storage* yaitu **local**.

PENGENALAN INFRASTRUCTURE AS CODE (IAC) |

APA ITU INFRASTRUCTURE AS CODE (IAC)?

- Menurut [Red Hat](#), **Infrastructure as Code (IaC)** adalah pengelolaan dan penyediaan infrastruktur melalui kode, bukan melalui proses manual.
- Melalui IaC, file konfigurasi yang dibuat memuat spesifikasi infrastruktur sehingga memudahkan penyesuaian dan pendistribusian konfigurasi serta memastikan penyediaan lingkungan yang sama setiap saat.
- Pemanfaatan IaC membantu dalam manajemen konfigurasi dan menghindari perubahan konfigurasi *ad-hoc* yang tidak terdokumentasi.
- Kontrol versi menjadi bagian penting dari IaC. File konfigurasi berada di bawah kontrol sumber seperti *file* kode sumber perangkat lunak lainnya.
- Menyebarkan infrastruktur sebagai kode dapat membagi infrastruktur menjadi komponen modular yang kemudian dapat digabungkan dengan berbagai cara melalui otomatisasi.

APA ITU INFRASTRUCTURE AS CODE (IAC)?

- Mengotomatisasi penyediaan infrastruktur dengan IaC membuat pengembang tidak perlu menyediakan dan mengelola server, sistem operasi, penyimpanan, dan komponen infrastruktur lainnya secara manual setiap kali mengembangkan atau menyebarkan aplikasi.
- Mengkodifikasi infrastruktur memberikan *template* yang harus diikuti untuk penyediaan, dan meskipun ini masih dapat dilakukan secara manual, alat otomatisasi, seperti Red Hat® Ansible® dapat membantu melakukan hal tersebut.

KEUNTUNGAN IAC

- a) Pengurangan biaya
- b) Peningkatan kecepatan penerapan
- c) Pengurangan kesalahan
- d) Peningkatan konsistensi infrastruktur
- e) Penghapusan penyimpangan konfigurasi

KONSEP DAN ARSITEKTUR ANSIBLE



APA ITU ANSIBLE?

- Menurut situs [Ansible](#), **Ansible** merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya.
- Menurut situs [Edureka](#), keuntungan menggunakan *Ansible* antara lain:
 1. **Simple**, menggunakan sintak penulisan yang ditulis menggunakan **YAML** yang disebut dengan **playbook**.
 2. **Agentless**, tidak diperlukan **agent** atau *software* khusus untuk diinstalasi pada host yang diautomasi.
 3. **Powerful & Flexible**, memiliki banyak modul untuk manajemen infrastruktur, jaringan, sistem operasi dan layanan.
 4. **Efficient**, tidak ada *software* yang diperlukan untuk instalasi pada server sehingga lebih banyak sumber daya yang dapat digunakan oleh aplikasi.

APA YANG DAPAT DILAKUKAN ANSIBLE?



1. Provisioning

Ansible memastikan paket-paket yang dibutuhkan akan diunduh dan diinstalasi sehingga aplikasi dapat diterapkan.

2. Configuration Management

Menetapkan dan mempertahankan konsistensi dari kinerja produk dengan mencatat dan memperbaharui informasi lengkap yang menjelaskan perangkat keras dan lunak perusahaan. Seperti versi dan pembaharuan yang telah diterapkan pada paket *software* yang terinstal dan lokasi serta alamat jaringan dari perangkat keras.

3. Application Deployment

Ansible dapat manajemen secara efektif keseluruhan *life cycle* dari aplikasi, mulai dari *development* sampai produksi.

4. Security and Compliance

Kebijakan keamanan dapat didefinisikan pada *Ansible* sehingga proses pemindaian dan pemulihan kebijakan ke lokasi dapat diintegrasikan ke dalam proses secara otomatis.



APA YANG DAPAT DILAKUKAN ANSIBLE?

5. Orchestration

Ansible menyediakan *orchestration* dalam arti menyelaraskan permintaan bisnis dengan aplikasi, data, dan infrastruktur. Ini mendefinisikan kebijakan dan tingkat layanan melalui alur kerja otomatis, penyediaan, dan manajemen perubahan sehingga menciptakan aplikasi selaras dengan infrastruktur yang dapat ditingkatkan dari atas ke bawah berdasarkan kebutuhan setiap aplikasi.



TERMINOLOGI PADA ANSIBLE

- **Controller Machine:** mesin dimana *ansible* diinstalasi dan bertanggungjawab untuk menjalankan *provisioning* pada *server* yang dikelola.
- **Inventory:** *file* inisialisasi yang memuat informasi tentang *server* yang dikelola.
- **Playbook:** *file* yang memuat kode *ansible* ditulis dalam format *YAML*. Playbook menjadi titik masuk untuk *Ansible provisioning*, dimana otomatisasi didefinisikan melalui tugas (*tasks*) menggunakan format *YAML*.
- **Task:** Blok yang mendefinisikan satu prosedur untuk dieksekusi, sebagai contoh instalasi *package* tertentu.
- **Module:** Modul merupakan abstraksi dari tugas sistem, seperti berkaitan dengan *package* atau membuat dan mengubah *file*. *Ansible* memiliki banyak modul *built-in* (bawaan), namun dapat juga dibuat modul khusus.

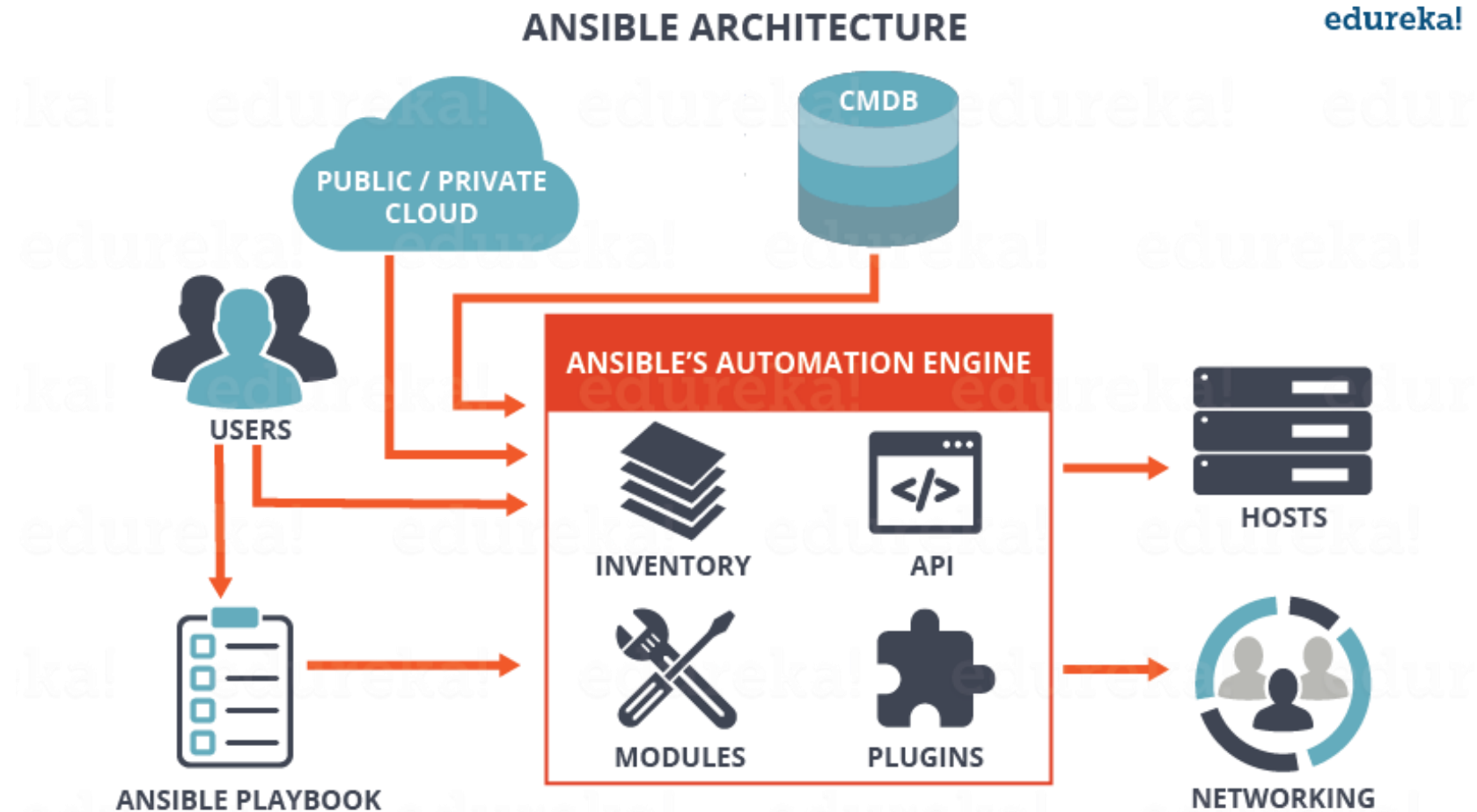


TERMINOLOGI PADA ANSIBLE

- **Role:** Cara yang telah ditentukan sebelumnya untuk mengatur *playbook* dan file lainnya untuk memfasilitasi berbagi pakai dan menggunakan kembali bagian dari *provisioning*.
- **Play:** *provisioning* yang dieksekusi mulai dari awal sampai akhir disebut dengan *play*. Dengan kata lain, eksekusi dari *playbook* disebut dengan *play*.
- **Facts:** *variable* global yang memuat informasi tentang sistem, seperti *interface* jaringan atau sistem operasi.
- **Handlers:** Digunakan untuk memicu perubahan status dari *service*, seperti *me-restart* atau menghentikan *service*.



ARSITEKTUR ANSIBLE



(Image source: <https://d1jnx9ba8s6j9r.cloudfront.net/blog/wp-content/uploads/2016/11/Ansible-Architecture-What-Is-Ansible-Edureka-768x449.png>)

ANSIBLE PROXMOX MODULE

1. `community.general.proxmox`

Digunakan untuk manajemen *instance* atau *container* di *Proxmox VE cluster* meliputi **create/delete/stop instance**. Kebutuhan untuk dapat menggunakan module ini adalah **proxmoxer** dan **requests** pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/community/general/proxmox_module.html

ANSIBLE PROXMOX MODULE

2. `community.general.proxmox_kvm`

Digunakan untuk manajemen **Qemu (Kernel-based Virtual Machine)** pada *Proxmox VE Cluster* meliputi **create/delete/stop Qemu (KVM) virtual machine**. Kebutuhan untuk dapat menggunakan module ini adalah **proxmoxer** dan **requests** pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/community/general/proxmox_kvm_module.html

ANSIBLE PROXMOX MODULE

3. **community.general.proxmox_template**

Digunakan untuk manajemen template dari sistem operasi pada *Proxmox VE cluster*. Kebutuhan untuk dapat menggunakan module ini adalah **proxmoxer** dan **requests** pada *host* yang mengeksekusi *module* tersebut. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/community/general/proxmox_template_module.html

ANSIBLE MODULE

4. **ansible.builtin.command**

Module ini dapat digunakan untuk mengeksekusi perintah-perintah pada target *host* termasuk utilitas **pvesh** dari PVE. *Module* ini dapat memfasilitasi manajemen objek pada PVE yang diotomasi menggunakan *Ansible* seperti *user*, *permission* dan lain sebagainya. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/command_module.html

ANSIBLE MODULE

5. **ansible.builtin.debug**

Module ini dapat digunakan untuk mencetak pernyataan saat eksekusi. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/debug_module.html

ANSIBLE MODULE

6. `ansible.builtin.ping`

Module ini dapat digunakan untuk mencoba untuk terhubung ke host, memverifikasi python yang dapat digunakan dan mengembalikan **pong** jika berhasil. Parameter-parameter yang dimiliki oleh *module* ini termasuk contoh penggunaannya dapat diakses pada alamat berikut:

https://docs.ansible.com/ansible/latest/collections/ansible/builtin/ping_module.html

3 (TIGA) CARA MENJALANKAN ANSIBLE



1. Ad-Hoc

```
ansible <inventory> -m
```

Sebagai contoh eksekusi *ansible* untuk *group inventory* dengan nama “**servers**” dan modul **ping**.

```
# ansible servers -m ping
```

2. Playbooks

```
ansible-playbook filename.yml
```

Sebagai contoh eksekusi *ansible* *playbook* dengan nama file **proxmox-iac.yml**.

```
# ansible-playbook proxmox-iac.yml
```

3. Automation Framework

Menurut situs [Ansible](#), **Ansible AWX** merupakan solusi berbasis web yang membuat *Ansible* lebih mudah digunakan untuk tim TI dan dirancang untuk menjadi pusat semua tugas otomasi serta memungkinkan untuk mengontrol akses pengguna. **Inventory** dapat dikelola secara grafis atau disinkronisasikan dengan beragam sumber **Cloud**. **AWX** juga mencatat (*log*) semua pekerjaan, terintegrasi dengan **LDAP** dan memiliki **API REST**. Tersedia pula tool **Command Line** untuk memudahkan integrasi dengan **Jenkins** juga.



INSTALASI DAN KONFIGURASI ANSIBLE CONTROL NODE

INSTALASI DAN KONFIGURASI ANSIBLE PADA CT 100 — ANSIBLE.BELAJAR.LOCAL

1. Memperbaharui *package index* dan melakukan *upgrade package* dari sistem **Ubuntu**.

```
apt update && apt upgrade -y
```

2. Mengatur **timezone** sistem.

```
timedatectl set-timezone Asia/Makassar
```

Sesuaikan nilai *timezone* dengan zona waktu yang digunakan.

3. Memverifikasi pengaturan *timezone*.

```
timedatectl
```

```
root@ansible:~# timedatectl
      Local time: Mon 2025-05-12 17:26:14 WITA
      Universal time: Mon 2025-05-12 09:26:14 UTC
          RTC time: n/a
          Time zone: Asia/Makassar (WITA, +0800)
System clock synchronized: yes
      NTP service: inactive
      RTC in local TZ: no
```

INSTALASI DAN KONFIGURASI ANSIBLE PADA CT 100 — ANSIBLE.BELAJAR.LOCAL

4. Menginstalasi *package* **software-properties-common** untuk menyediakan abstraksi terkait penggunaan *repository apt*.

```
apt -y install software-properties-common
```

5. Menambahkan *repository* **Personal Package Archive (PPA) Ansible**.

```
apt-add-repository ppa:ansible/ansible
```

Tekan **Enter** pada pesan konfirmasi penambahan *repository* yang tampil

6. Memperbaharui *package index*.

```
apt update
```

7. Menginstalasi **Ansible** menggunakan **PPA**.

```
apt -y install ansible
```

INSTALASI DAN KONFIGURASI ANSIBLE PADA CT 100 — ANSIBLE.BELAJAR.LOCAL

8. Menampilkan versi **Ansible** yang terinstalasi.

`ansible --version`

```
root@ansible:~# ansible --version
ansible [core 2.18.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible
/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collection
s
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
```

MENGATUR PASSWORDLESS SECURE SHELL (SSH) LOGIN

Membuat **key pair** agar **CT 100-ansible.belajar.local** sebagai **control node** dapat mengakses *managed machine* yaitu **Server PVE** melalui SSH tanpa *password*.

ssh-keygen

Tampil inputan “**Enter file in which to save the key (/root/.ssh/id_ed25519):**”

```
root@ansible:~# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
```

Tekan **Enter** untuk menggunakan lokasi dan nama *file default* sebagai penyimpanan *key* yaitu **/root/.ssh/id_ed25519**.

Selanjutnya tekan **Enter** kembali pada inputan “**Enter passphrase (empty for no passphrase):**” dan “**Enter same passphrase again:**” untuk mengosongkan *passphrase*.

MENGATUR PASSWORDLESS SECURE SHELL (SSH) LOGIN

Terlihat proses pembuatan key. Tunggu hingga proses selesai dilakukan.

```
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /root/.ssh/id_ed25519  
Your public key has been saved in /root/.ssh/id_ed25519.pub  
The key fingerprint is:  
SHA256:/QeYOpwmWS0jUHAos/3tzKs5ZgoSNOB014d4I89EkY root@ansible  
The key's randomart image is:  
+--[ED25519 256]--+  
|    ....E.    |  
|.  ..O  .    |  
|..O .+.  .    |  
|.O O.==. . O  |  
| o+ *O.+S + . |  
| oo+...=.O . . |  
|o ...O= *    . |  
|o  . *=O..    . |  
| .. +..+..    |  
+-----[SHA256]-----+
```

MENGATUR PASSWORDLESS SECURE SHELL (SSH) LOGIN

Menyalinkan **public key** dari **CT Ansible Control Node** ke **managed machine** yaitu **Server PVE**.

```
ssh-copy-id -i .ssh/id_ed25519.pub root@192.168.0.1
```

Ketik **yes** dan tekan **Enter** pada pesan konfirmasi “**Are you sure you want to continue connection (yes/no)?**”.

```
root@ansible:~# ssh-copy-id -i .ssh/id_ed25519.pub root@192.168.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_ed25519.pub"
The authenticity of host '192.168.0.1 (192.168.0.1)' can't be established.
ED25519 key fingerprint is SHA256:RwGty0q8xybiohRXYJi2qRcUtjqaiMcOfn/bh2UDbMU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Selanjutnya tampil inputan **password** untuk *user* **root** dari *managed machine* **Server PVE** dengan alamat IP **192.168.0.1**. Masukkan sandinya dan tekan **Enter**.

MENGATUR PASSWORDLESS SECURE SHELL (SSH) LOGIN

Proses penyalinan *public* key berhasil dilakukan.

```
root@ansible:~# ssh-copy-id -i .ssh/id_ed25519.pub root@192.168.0.1
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: ".ssh/id_ed25519.pub"
The authenticity of host '192.168.0.1 (192.168.0.1)' can't be established.
ED25519 key fingerprint is SHA256:RwGty0q8xybiohRXYJi2qRcUtjqaiMcOfn/bh2UDbMU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now i
t is to install the new keys
root@192.168.0.1's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@192.168.0.1'"
and check to make sure that only the key(s) you wanted were added.

root@ansible:~#
```

MENGATUR PASSWORDLESS SECURE SHELL (SSH) LOGIN

Memverifikasi hasil pengaturan *passwordless* SSH dengan cara mengakses *managed machine* yaitu **server PVE** menggunakan *SSH Client*.

```
ssh root@192.168.0.1
```

```
root@ansible:~# ssh root@192.168.0.1

Panduan: https://idcloudhost.com/panduan
=====
- Hostname.....: pve
- Disk Space.....: 51G113M remaining
=====
- CPU usage.....: 0.02, 0.06, 0.02 (1, 5, 15 min)
- Memory used.....: 1461 MB / 3918 MB
- Swap in use.....: 0 MB
=====

Linux pve 6.8.12-10-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-10 (2025-04-18T07:39Z) 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@pve:~#
```

MENGINSTALASI PAKET PYTHON3-PIP DAN PROXMOXER PADA PVE

- Menginstalasi paket **python3-pip**.

```
apt -y install python3-pip
```

- Menginstalasi paket **proxmoxer** menggunakan **pip3**.

```
pip3 install proxmoxer --break-system-packages
```

```
root@pve:~# pip3 install proxmoxer --break-system-packages
Collecting proxmoxer
  Downloading proxmoxer-2.2.0-py3-none-any.whl (19 kB)
Installing collected packages: proxmoxer
Successfully installed proxmoxer-2.2.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting
behaviour with the system package manager. It is recommended to use a virtual environme
nt instead: https://pip.pypa.io/warnings/venv
```

MEMBUAT USER BERTIPE PAM UNTUK DIGUNAKAN OLEH ANSIBLE PLAYBOOK

Terdapat 3 (tiga) tahapan untuk menambahkan *user* bertipe **Pluggable Authentication Modules (PAM)** di PVE yaitu:

1. Membuat *user* di sistem operasi.

adduser ansible

```
root@pve:~# adduser ansible
Adding user `ansible' ...
Adding new group `ansible' (1002) ...
Adding new user `ansible' (1002) with group `ansible (1002)' ...
Creating home directory `/home/ansible' ...
Copying files from `/etc/skel' ...
New password: _____
Retype new password: _____
passwd: password updated successfully
Changing the user information for ansible
Enter the new value, or press ENTER for the default
    Full Name []: _____
    Room Number []: _____
    Work Phone []: _____
    Home Phone []: _____
    Other []: _____
Is the information correct? [Y/n] y
Adding new user `ansible' to supplemental / extra groups `users' ...
Adding user `ansible' to group `users' ...
```

Pada inputan **New password** dan **Retype new password**, masukkan sandi yang digunakan oleh *user* **ansible**. Sedangkan pada inputan **Full Name []** hingga **Other []**, tekan **Enter** untuk mengosongkan isian parameter tersebut dan melanjutkan ke proses berikutnya. Terakhir pada inputan **Is the information correct? [Y/n]**, ketik **y**.

MEMBUAT USER BERTIPE PAM UNTUK DIGUNAKAN OLEH ANSIBLE PLAYBOOK

2. Menambahkan *user* di sistem operasi ke PVE dan memverifikasi hasil penambahan *user* tersebut.

```
pveum user add ansible@pam
```

```
pveum user list
```

```
root@pve:~# pveum user add ansible@pam
root@pve:~# pveum user list
```

userid	comment	email	enable	expire	firstname	groups	keys	lastname
ansible@pam			1	0				
root@pam			1	0				

MEMBUAT USER BERTIPE PAM UNTUK DIGUNAKAN OLEH ANSIBLE PLAYBOOK

3. Mengatur ijin akses (**permission**) menggunakan *roles* **Administrator** ke user **ansible@pam** terhadap **object** dengan **path** / (seluruh objek) dan memverifikasi hasil pengaturan ijin akses tersebut.

```
pveum acl modify / --roles Administrator --user ansible@pam
```

```
pveum acl list
```

```
root@pve:~# pveum acl modify / --roles Administrator --user ansible@pam
root@pve:~# pveum acl list
```

path	roleid	type	ugid	propagate
/	Administrator	user	ansible@pam	1

MEMBUAT USER BERTIPE PAM UNTUK DIGUNAKAN OLEH ANSIBLE PLAYBOOK

Keluar dari akses SSH Server PVE.

`exit`

```
root@pve:~# exit
logout
Connection to 192.168.0.1 closed.
root@ansible:~#
```

IMPLEMENTASI DASAR ANSIBLE

MENGATUR ANSIBLE INVENTORY

- **Inventory** merupakan file inisialisasi yang digunakan oleh *Ansible* untuk mendaftarkan dan mengelompokkan mesin atau *host* yang akan dikelola.
- Membuat direktori bernama **belajar** sebagai lokasi *file-file* proyek *Ansible* dan berpindah ke direktori tersebut.

```
mkdir belajar
```

```
cd belajar
```

```
root@ansible:~# mkdir belajar
root@ansible:~# cd belajar
root@ansible:~/belajar#
```

MENGATUR ANSIBLE INVENTORY

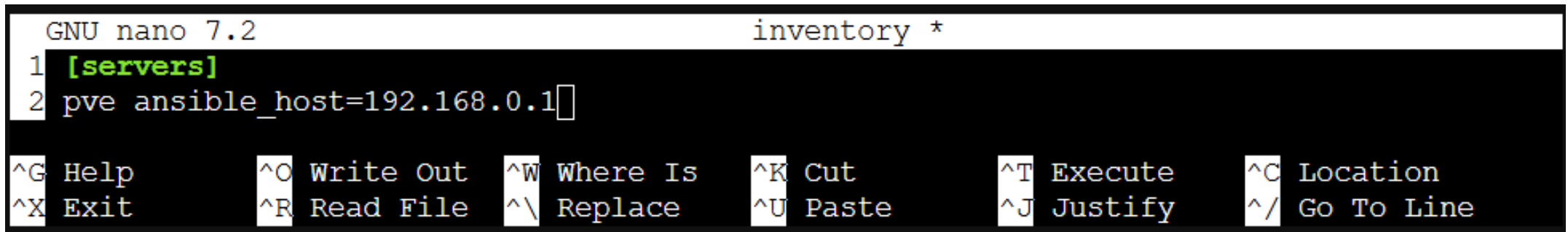
Menambahkan **managed machines** ke file *inventory* menggunakan editor **nano**.

`nano -l inventory`

Konten dari *file* tersebut adalah sebagai berikut:

`[servers]`

`pve ansible_host=192.168.0.1`

A screenshot of the GNU nano 7.2 text editor. The title bar at the top shows "GNU nano 7.2" on the left and "inventory *" on the right. The editor area has a black background with white text. Line 1 contains "[servers]" in green. Line 2 contains "pve ansible_host=192.168.0.1" followed by a white cursor. At the bottom, there is a status bar with various keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^C Location, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, ^J Justify, and ^/ Go To Line.

```
GNU nano 7.2                                inventory *
1  [servers]
2  pve ansible_host=192.168.0.1
^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
```

Variable **ansible_host** digunakan untuk mengatur alamat IP dari target atau *managed machine* yaitu dalam hal ini adalah alamat IP dari setiap *server*. Sebagai contoh **PVE** merupakan *inventory_hostname* atau *alias* bagi *server* dengan alamat **192.168.0.1**. Simpan perubahan dengan menekan **CTRL+O** dan tekan **Enter**. Keluar dari editor *nano* dengan menekan **CTRL+X**.

MENGATUR ANSIBLE.CFG

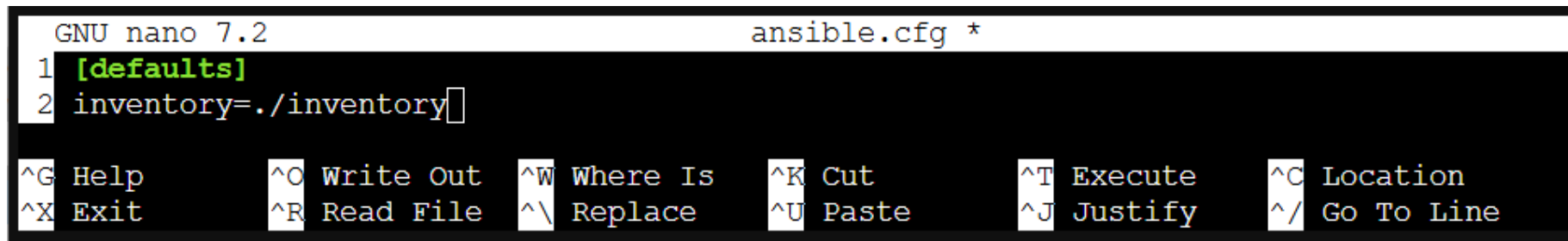
File **ansible.cfg** memuat konfigurasi perilaku dari *Ansible*

```
nano -l ansible.cfg
```

Konten dari *file* tersebut adalah sebagai berikut:

```
[defaults]
```

```
inventory=./inventory
```



```
GNU nano 7.2                               ansible.cfg *
1 [defaults]
2 inventory=./inventory
^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
```

Pada *section defaults* dilakukan pengaturan parameter **inventory=./inventory** yang digunakan untuk menentukan *path* atau lokasi dari *file inventory* berada yaitu **./inventory**. Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**. Keluar dari editor **nano** dengan menekan tombol **CTRL+X**.

ANSIBLE AD-HOC MODULE PING

Memverifikasi koneksi ke seluruh **managed machines** untuk group “**servers**” menggunakan modul **ping**.

```
ansible servers -m ping
```

```
root@ansible:~/belajar# ansible servers -m ping
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | SUCCESS => {
  "ansible_facts": {
    "discovered_interpreter_python": "/usr/bin/python3.11"
  },
  "changed": false,
  "ping": "pong"
}
```

Output menunjukkan bahwa verifikasi koneksi ke host **pve** yang tergabung dalam group **servers** berhasil dilakukan dimana ditandai dengan pesan **pong**. Selain itu terlihat pula informasi **interpreter python** ditemukan pada host **pve** yaitu berada di **/usr/bin/python3.11**.

ANSIBLE AD-HOC MODULE COMMAND

Menampilkan informasi *system uptime* dari *managed machine* dengan nama *group servers* menggunakan *module command*.

```
ansible servers -m command -a "uptime"
```

```
root@ansible:~/belajar# ansible servers -m command -a "uptime"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
18:27:04 up 2:17, 1 user, load average: 0.00, 0.00, 0.00
```

Option -a digunakan untuk mencantumkan argumen.

Output menunjukkan bahwa *host pve* yang tergabung dalam *group servers* telah aktif selama 2 jam 17 menit dan terdapat satu pengguna yang sedang aktif.

MEMBUAT ANSIBLE PLAYBOOK SEDERHANA

```
nano -l playbook.yml
```

Konten dari *file* tersebut adalah sebagai berikut:

```
---
```

```
- name: Belajar
```

```
  hosts: servers
```

```
  vars:
```

```
    training: "Proxmox IaC"
```

```
  tasks:
```

```
    - name: Menampilkan isi variable
```

```
      ansible.builtin.debug:
```

```
        msg: "{{ training }}"
```

MEMBUAT ANSIBLE PLAYBOOK SEDERHANA

```
GNU nano 7.2                                playbook.yml
1  ---
2  - name: Belajar
3    hosts: servers
4    vars:
5      training: "Proxmox IaC"
6    tasks:
7      - name: Menampilkan isi variable
8        ansible.builtin.debug:
9          msg: "{{ training }}"
10
[ Read 9 lines ]
^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
```

- Playbooks memuat Plays
 - Plays memuat Tasks
 - Tasks memanggil Modules

MEMBUAT ANSIBLE PLAYBOOK SEDERHANA

Penjelasan:

- Baris 1: `---` (3 hyphen) merupakan awal dari dokumen YAML atau *playbook*.
- Baris 2: **name** digunakan untuk menentukan nama dari **Play** yaitu “**Belajar**”.
- Baris 3: **hosts** digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi task yaitu grup **servers**.
- Baris 4: **vars** digunakan untuk mendefinisikan *variable* pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 5: Deklarasi *variable* dengan nama “**training**” yang memiliki nilai “**Proxmox IaC**”.
- Baris 6: **tasks** merupakan daftar tindakan atau aksi yang perlu dieksekusi.

MEMBUAT ANSIBLE PLAYBOOK SEDERHANA

- Baris 7: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menampilkan isi variable**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 8: nama *module* yang dieksekusi yaitu **ansible.builtin.debug** yang berguna untuk menampilkan pesan ketika eksekusi. Modul ini memiliki parameter **msg** yang digunakan untuk menampilkan pesan tertentu termasuk isi dari *variable*. *Variabel* direferensikan dengan menempatkan nama *variabel* dalam kurung kurawal ganda (**{{ }}**). Sebagai contoh **{{ training }}**

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari editor **nano** dengan menekan tombol **CTRL+X**.

MENGECEK SINTAK PLAYBOOK SEBELUM DIEKSEKUSI

```
ansible-playbook playbook --syntax-check
```

```
root@ansible:~/belajar# ansible-playbook playbook.yml --syntax-check  
playbook: playbook.yml  
root@ansible:~/belajar#
```

Pengecekan sintak dari *playbook* dilakukan sebelum dieksekusi agar memastikan tidak terdapat permasalahan terkait sintak.

MENAMPILKAN INFORMASI HOST TERDAMPAK OLEH PLAYBOOK SEBELUM DIEKSEKUSI

```
ansible-playbook playbook.yml --list-hosts
```

```
root@ansible:~/belajar# ansible-playbook playbook.yml --list-hosts
playbook: playbook.yml

play #1 (servers): Belajar    TAGS: []
  pattern: ['servers']
  hosts (1):
    pve
```

Terlihat bahwa *host* **pve** pada grup **servers** yang terdampak oleh eksekusi *playbook*.

MENGEKSEKUSI PLAYBOOK

ansible-playbook playbook.yml

```
root@ansible:~/belajar# ansible-playbook playbook.yml

PLAY [Belajar] 1 *****

TASK [Gathering Facts] 2 *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [Menampilkan isi variable] 3 *****
ok: [pve] => {
  "msg": "Proxmox IaC" 4
}

PLAY RECAP *****
pve : ok=2 5 changed=0 unreachable=0 failed=0 skipped=0
    rescued=0 ignored=0
```

MENGEKSEKUSI PLAYBOOK

Penjelasan:

- (1) Nama **play** yang dieksekusi yaitu “**Belajar**”
- (2) Eksekusi **task Gathering Facts** yang memuat informasi terkait *remote host* dan terlihat pesan status **ok** yang menyatakan *task* dapat dieksekusi dengan baik
- (3) Eksekusi **task “Menampilkan isi variable”** dan terlihat pesan status **ok** yang menyatakan *task* dapat dieksekusi dengan baik
- (4) Menampilkan isi dari *variable training* menggunakan module **ansible.builtin.debug** yaitu “**Proxmox laC**”.
- (5) *Metric ok* menginformasikan rangkuman jumlah *task* yang dapat dieksekusi dengan baik atau tidak mengubah apapun pada **servers**.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Mengunduh *container template* alpine versi 3.21.

```
nano -l unduh-container-template.yml
```

Konten dari *file* tersebut adalah sebagai berikut:

```
---
```

```
- name: Manajemen Container Template di PVE
```

```
  hosts: servers
```

```
  tasks:
```

```
    - name: Unduh Container Template
```

```
      community.general.proxmox_template:
```

```
        node: pve
```

```
        api_user: ansible@pam
```

```
        api_password: Ansible2025!
```

PENGGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

api_host: pve

storage: local

content_type: vztpl

template: alpine-3.21-default_20241217_amd64.tar.xz

timeout: 60

```
GNU nano 7.2 unduh-container-template.yml
1 ---
2 - name: Manajemen Container Template di PVE
3   hosts: servers
4   tasks:
5   - name: Unduh Container Template
6     community.general.proxmox_template:
7       api_host: pve
8       api_user: ansible@pam
9       api_password: Ansible2025!
10      node: pve
11      storage: local
12      content_type: vztpl
13      template: alpine-3.21-default_20241217_amd64.tar.xz
14      timeout: 60
15
```

[Read 14 lines]

^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

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Penjelasan:

- Baris 1: `---` (3 hyphen) merupakan awal dari dokumen YAML atau *playbook*.
- Baris 2: **name** digunakan untuk menentukan nama dari *play* yaitu “**Manajemen Container Template di PVE**”.
- Baris 3: **hosts** digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu grup **servers**.
- Baris 4: **tasks** merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 5: **name** digunakan untuk menentukan nama dari *task* yaitu “**Unduh Container Template**”. Di dalam setiap task memuat kode yang berhubungan dengan modul yang harus dieksekusi.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

- Baris 6: nama modul yang dieksekusi yaitu **community.general.proxmox_template**.
- Baris 7: **api_host** merupakan parameter dari *module proxmox_template* untuk menentukan *host* dari PVE yaitu “**pve**”.
- Baris 8: **api_user** merupakan parameter dari *module proxmox_template* untuk menentukan *user* yang digunakan login ke PVE yaitu “**ansible@pam**”
- Baris 9: **api_password** merupakan parameter dari *module proxmox_template* untuk menentukan password otentikasi ke PVE yaitu “**Ansible2025!**”.
- Baris 10: **node** merupakan parameter dari *module proxmox_template* untuk menentukan node dari PVE sebagai lokasi *container template* yaitu **pve**.
- Baris 11: **storage** merupakan parameter dari *module proxmox_template* untuk menentukan lokasi penyimpanan *container template* yang diunduh yaitu **local**.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

- Baris 12: **content_type** merupakan parameter dari *module proxmox_template* untuk menentukan jenis konten yaitu **vztmpl**.
- Baris 13: **template** merupakan parameter dari *module proxmox_template* untuk menentukan nama dari *file container template* yang akan diunduh “**alpine-3.21-default_20241217_amd64.tar.xz**”.
- Baris 14: **timeout** merupakan parameter dari *module proxmox_template* untuk mengatur batas waktu dari operasi yaitu 60 detik.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Memverifikasi sintak dari *playbook*.

```
ansible-playbook unduh-container-template.yml --syntax-check
```

Mengeksekusi *playbook*.

```
ansible-playbook unduh-container-template.yml
```

```
root@ansible:~/belajar# ansible-playbook unduh-container-template.yml

PLAY [Manajemen Container Template di PVE] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [Unduh Container Template] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    skipped=0
rescued=0    ignored=0
```

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Memverifikasi *container template* **alpine 3.21** telah berhasil diunduh.

```
ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

```
root@ansible:~/belajar# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

format	size	valid	ctime	encrypted	notes	parent	protected	used	verification	vmid
raw	4.00 GiB	local:100/vm-100-disk-0.raw	1747480046					2.08 GiB		100
txz	3.01 MiB	local:vztmpl/alpine-3.21-default_20241217_amd64.tar.xz	1747478885							
tzst	135.03 MiB	local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst	1747022586							

Terlihat *container template* **alpine 3.21** telah berhasil diunduh.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Membuat *playbook* untuk menghapus *container template* **alpine 3.21**.

```
nano -l hapus-container-template.yml
```

Konten dari *file* tersebut adalah sebagai berikut:

```
---
```

```
- name: Manajemen Container Template di PVE
```

```
  hosts: servers
```

```
  vars:
```

```
    api_host: pve
```

```
    api_user: ansible@pam
```

```
    api_password: Ansible2025!
```

PENGGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

```
node: pve
```

```
ct_ostemplate: alpine-3.21-default_20241217_amd64.tar.xz
```

```
tasks:
```

- name: Hapus Container Template

```
community.general.proxmox_template:
```

```
  api_host: "{{ api_host }}"
```

```
  api_user: "{{ api_user }}"
```

```
  api_password: "{{ api_password }}"
```

```
  node: "{{ node }}"
```

```
  template: "{{ ct_ostemplate }}"
```

```
  state: absent
```

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

```
GNU nano 7.2 hapus-container-template.yml
1 ---
2 - name: Manajemen Container Template di PVE
3   hosts: servers
4   vars:
5     api_host: pve
6     api_user: ansible@pam
7     api_password: Ansible2025!
8     node: pve
9     ct_ostemplate: alpine-3.21-default_20241217_amd64.tar.xz
10  tasks:
11    - name: Hapus Container Template
12      community.general.proxmox_template:
13        api_host: "{{ api_host }}"
14        api_user: "{{ api_user }}"
15        api_password: "{{ api_password }}"
16        node: "{{ node }}"
17        template: "{{ ct_ostemplate }}"
18        state: absent
19
```

^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

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Penjelasan:

- Baris 1: `---` (3 hyphen) merupakan awal dari dokumen YAML atau *playbook*.
- Baris 2: **name** digunakan untuk menentukan nama dari *play* yaitu **“Manajemen Container Template di PVE”**.
- Baris 3: **hosts** digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu grup **servers**.
- Baris 4: **vars** digunakan untuk mendefinisikan variable pada *playbook* dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.
- Baris 5-9: Deklarasi 5 (lima) **variable** yaitu **“api_host”** dengan nilai **“pve”**, **“api_user”** dengan nilai **“ansible@pam”**, **“api_password”** dengan nilai **“Ansible2025!”** dan **“node”** dengan nilai **“pve”** serta **“ct_ostemplate”** dengan nilai **“alpine-3.21-default_20241217_amd64.tar.xz”**.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

- Baris 10: **tasks** merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 11: **name** digunakan untuk menentukan nama dari *task* yaitu “**Unduh Container Template**”. Di dalam setiap task memuat kode yang berhubungan dengan modul yang harus dieksekusi.
- Baris 12: nama modul yang dieksekusi yaitu **community.general.proxmox_template**.
- Baris 13: **api_host** merupakan parameter dari *module proxmox_template* untuk menentukan *host* dari PVE yang diambil dari *variable api_host*.
- Baris 14: **api_user** merupakan parameter dari *module proxmox_template* untuk menentukan *user* yang digunakan login ke PVE yang diambil dari *variable api_user*.
- Baris 15: **api_password** merupakan parameter dari *module proxmox_template* untuk menentukan password otentikasi ke PVE yang diambil dari nilai *variable api_password*.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

- Baris 16: **node** merupakan parameter dari *module proxmox_template* untuk menentukan node dari PVE sebagai lokasi *container template* yang diambil dari nilai *variable node*.
- Baris 17: **template** merupakan parameter dari *module proxmox_template* untuk menentukan nama dari *file container template* yang akan dihapus yang diambil dari nilai *variable ct_ostemplate*.
- Baris 18: **state** merupakan parameter dari *module proxmox_template* untuk menentukan status *template* yang diinginkan yaitu menghapus *container template* dengan nilai **absent**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Memverifikasi sintak dari *playbook*.

```
ansible-playbook hapus-container-template.yml --syntax-check
```

Mengeksekusi *playbook*.

```
ansible-playbook hapus-container-template.yml
```

```
root@ansible:~/belajar# ansible-playbook hapus-container-template.yml

PLAY [Manajemen Container Template di PVE] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [Hapus Container Template] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=2    changed=1    unreachable=0    failed=0    skipped=0
                    rescued=0    ignored=0
```

PENGUNAAN MODUL ANSIBLE PROXMOX_TEMPLATE

Memverifikasi *container template alpine 3.21* telah dihapus.

```
ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
```

```
root@ansible:~/belajar# ansible servers -m command -a "pvesh get /nodes/pve/storage/local/content"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

format	size	valid	ctime	encrypted	notes	parent	protected	used	verification	vmid
raw	4.00 GiB	local:100/vm-100-disk-0.raw	1747480598					2.08 GiB		100
tzst	135.03 MiB	local:vztmpl/ubuntu-24.04-standard_24.04-2_amd64.tar.zst	1747022586							

Tidak terlihat *container template alpine 3.21*.

STUDI KASUS

INSTALASI PAKET GIT DAN TREE

- Menginstalasi paket **git** yang diperlukan untuk melakukan **clone repository Proxmox IAC** dari **Github**.

```
apt -y install git
```

- Menginstalasi paket **tree** yang diperlukan untuk menampilkan konten dari direktori secara rekursif sehingga terlihat struktur hirarkinya.

```
apt -y install tree
```

CLONE REPOSITORY

- Melakukan *clone repository* **proxmox-iac** dari **Github Repository I Putu Hariyadi**.

```
git clone https://github.com/iputuhariyadi/proxmox-iac.git
```

- Berpindah ke direktori **proxmox-iac**

```
cd proxmox-iac
```

- Menampilkan isi dari direktori **proxmox-iac** secara rekursif sehingga terlihat struktur hirarki konten didalamnya.

```
tree
```

```
root@ansible:~/proxmox-iac# tree
.
|-- ansible.cfg
|-- data_pengguna.yml
|-- inventory
|-- membuat-container.yml
|-- membuat-permission.yml
|-- membuat-user.yml
|-- menghapus-container.yml
|-- menghapus-permission.yml
|-- menghapus-user.yml
|-- power-management-container.yml
|-- proxmox-iac.yml
`-- pve_auth_vars.yml

1 directory, 12 files
```

MENGATUR INVENTORY

Membuka *file* bernama **inventory** yang memuat daftar *host* yang dikelola.

```
nano -l inventory
```

```
GNU nano 7.2                                inventory
1  [servers]
2  pve ansible_host=192.168.0.1
3
^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
```

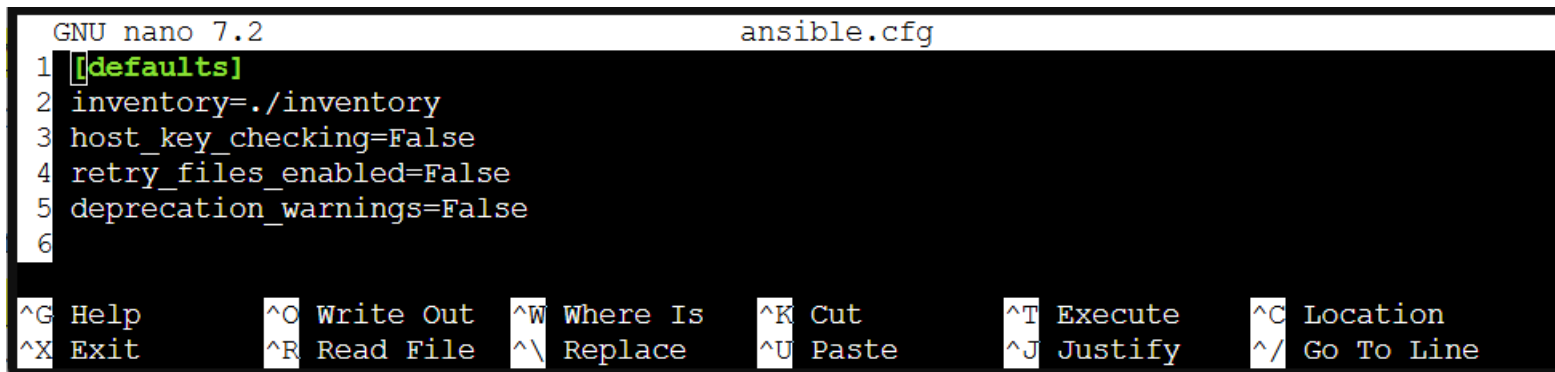
Terlihat terdapat satu grup bernama **servers** dengan satu *host* anggota yaitu **pve**. **pve** merupakan **inventory_hostname** atau nama **alias** bagi *host* dengan alamat IP **192.168.0.1** yang menjadi nilai dari *variable* **ansible_host**. **ansible_host** merupakan *variable* yang digunakan oleh *Ansible* untuk melakukan koneksi dengan *remote host* alias **pve**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MENGATUR FILE ANSIBLE.CFG

Membuka *file* bernama **ansible.cfg** yang memuat konfigurasi perilaku dari *Ansible*.

```
nano -l ansible.cfg
```



```
GNU nano 7.2 ansible.cfg
1 [defaults]
2 inventory=./inventory
3 host_key_checking=False
4 retry_files_enabled=False
5 deprecation_warnings=False
6
^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
```

Pada *section defaults* terdapat 4 (empat) parameter yang diatur yaitu:

- **inventory=./inventory** digunakan untuk menentukan *path* atau lokasi dari *file inventory* berada yaitu *./inventory*.
- **host_key_checking=False** digunakan untuk menonaktifkan pengecekan SSH *host key*.

MENGATUR FILE ANSIBLE.CFG

- **retry_files_enabled=False** digunakan untuk menonaktifkan pembuatan file `.retry` saat playbook mengalami kegagalan.
- **deprecation_warnings=False** digunakan untuk menonaktifkan peringatan penghentian.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE VARIABLE PVE_AUTH_VARS.YML

File **pve_auth_vars.yml** memuat *variable-variable* yang diperlukan untuk melakukan koneksi ke **Server PVE** dari *Ansible Playbook*.

```
nano -l pve_auth_vars.yml
```

```
GNU nano 7.2                                pve_auth_vars.yml
1 api_host: pve
2 api_user: ansible@pam
3 api_password: Ansible2025!
4 node: pve
5 proxmox_api_url: https://192.168.0.1:8006/api2/json
6

^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
```

Terdapat 5 (lima) variable yang dideklarasikan yaitu:

- Variable **api_host** digunakan untuk menampung nama *host* dari **Server PVE** yaitu **pve**.

MEMBUAT FILE VARIABLE PVE_AUTH_VARS.YML

- Variable **api_user** digunakan untuk menampung nama *login user* ke **Server PVE** yaitu **ansible@pam**.
- Variable **api_password** digunakan untuk menampung sandi *login* dari *user* yang digunakan untuk terkoneksi ke **Server PVE**, sebagai contoh **Ansible2025!**. **Jika berbeda maka ganti dengan sandi yang digunakan.**
- Variable **node** digunakan untuk menampung nama *node* dari **Server PVE** yaitu **pve**.
- Variable **proxmox_api_url** digunakan untuk menampung alamat **Uniform Resource Locator (URL)** dari **Application Programming Interface (API)** *Server PVE* yaitu **https://192.168.0.1:8006/api2/json**

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE VARIABLE DATA_PENGGUNA.YML

File **data_pengguna.yml** memuat *variable-variable* yang diperlukan untuk melakukan pembuatan **Container (CT)**, **Users** dan **Permissions** menggunakan *Ansible Playbook*.

```
nano -l data_pengguna.yml
```

```
GNU nano 7.2
```

```
1 ---
2 ct_root_password: mandalika
3 ct_ostemplate: ubuntu-24.04-standard_24.04-2_amd64.tar.zst
4 ct_nameserver: 1.1.1.1
5 ct_searchdomain: "belajar.local"
6 ct_gateway: 192.168.0.1
7 user_password: bumigora
8 proles: PVEVMUser
```

Penjelasan:

- Baris 1: --- (3 hyphen) merupakan awal dari dokumen YAML.
- Baris 2: deklarasi *variable* bernama **ct_root_password** dengan nilai **mandalika** yang digunakan untuk mengatur *password* atau sandi dari *user* ketika pembuatan *container*.

MEMBUAT FILE VARIABLE DATA_PENGGUNA.YML

- Baris 3: Deklarasi *variable* bernama **ct_ostemplate** dengan nilai **ubuntu-24.04-standard_24.04-2_amd64.tar.zst** yang digunakan untuk *password* atau sandi ketika pembuatan **user**.
- Baris 4: Deklarasi *variable* bernama **ct_nameserver** dengan nilai **1.1.1.1** yang digunakan untuk pengaturan **Server DNS** ketika pembuatan **container**.
- Baris 5: Deklarasi *variable* bernama **ct_search_domain** dengan nilai **belajar.local** yang digunakan untuk pengaturan **domain DNS** ketika pembuatan **container**.
- Baris 6: Deklarasi *variable* bernama **ct_gateway** dengan nilai **192.168.0.1** yang digunakan untuk pengaturan **IP Gateway** ketika pembuatan **container**.
- Baris 7: Deklarasi *variable* bernama **user_password** dengan nilai **bumigora** yang digunakan untuk *password* atau sandi ketika pembuatan **user**.
- Baris 8: Deklarasi *variable* bernama **proles** dengan nilai **PVEVMUser** yang digunakan untuk mengatur *role* dari *user* ketika pengaturan **permission** atau ijin akses terhadap objek di PVE.

MEMBUAT FILE VARIABLE DATA_PENGGUNA.YML

```
9 siswa:
10   - nis: 2501
11     nama_depan: Jorge
12     nama_belakang: Martin
13     container:
14       vmid: 101
15       hostname: server101.belajar.local
16       ip_address: 192.168.0.101/24
17   - nis: 2502
18     nama_depan: Francesco
19     nama_belakang: Bagnaia
20     container:
21       vmid: 102
22       hostname: server102.belajar.local
23       ip_address: 192.168.0.102/24
```

```
24   - nis: 2503
25     nama_depan: Marc
26     nama_belakang: Marquez
27     container:
28       vmid: 103
29       hostname: server103.belajar.local
30       ip_address: 192.168.0.103/24
31   - nis: 2504
32     nama_depan: Ai
33     nama_belakang: Ogura
34     container:
35       vmid: 104
36       hostname: server104.belajar.local
37       ip_address: 192.168.0.104/24
```

MEMBUAT FILE VARIABLE DATA_PENGGUNA.YML

Baris 9-37 merupakan deklarasi *variable* bernama **siswa** dengan struktur data **list of dictionaries** yang didalamnya memuat 4 (empat) data untuk pembuatan **container, user** dan pengaturan **permission** (ijin akses) terhadap objek di PVE, seperti terlihat pada tabel berikut:

nis	nama_depan	nama_belakang	container		
			vmid	hostname	ip_address
2501	Jorge	Martin	101	server101.belajar.local	192.168.0.101/24
2502	Francesco	Bagnaia	102	server102.belajar.local	192.168.0.102/24
2503	Marc	Marquez	103	server103.belajar.local	192.168.0.103/24
2504	Ai	Ogura	104	server104.belajar.local	192.168.0.104/24

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOT-IAC.YML

File **proxmot-iac.yml** merupakan playbook yang didalamnya memuat *task-task* untuk membuat dan menghapus **Container (CT)**, **Users** dan **Permissions** serta melakukan *power management* bagi **CT** meliputi menjalankan (**started**), menjalankan ulang (**restarted**), menghentikan (**stopped**).

```
nano -l proxmot-iac.yml
```

```
GNU nano 7.2 proxmot-iac.yml
1  --
2  - name: Proxmox IaC
3    hosts: servers
4    vars:
5      aksi: "buat"
6    vars_files:
7      - data_pengguna.yml
8    tasks:
9      - name: include vars
10        include_vars: pve_auth_vars.yml
11        no_log: true
```

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

Baris 1: `---` (3 hyphen) merupakan awal dari dokumen YAML.

Baris 2: **name** digunakan untuk menentukan nama dari *Play* yaitu "**Proxmox IaC**".

Baris 3: **hosts** digunakan untuk menentukan daftar *host* atau *group host* sebagai lokasi eksekusi *task* yaitu grup **servers**.

Baris 4: **vars** digunakan untuk mendefinisikan *variable* pada *playbook*.

Baris 5: Deklarasi *variable* bernama **aksi** dengan nilai "**buat**" dimana penggunaannya serupa dengan yang terdapat di bahasa pemrograman.

Baris 6-7: **vars_files** digunakan untuk menyisipkan secara eksplisit *file variable* dengan nama **data_pengguna.yml**.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 8: **tasks** merupakan daftar tindakan atau aksi yang perlu dieksekusi.
- Baris 9: **name** digunakan untuk menentukan nama dari *task* yaitu “**include vars**”. Di dalam setiap *task* memuat kode yang berhubungan dengan modul yang harus dieksekusi.
- Baris 10: **include_vars** digunakan untuk memuat *variable* dari *file* yaitu “**pve_auth_vars.yml**”. *Variable* di dalam *file* tersebut diperlukan untuk pembentukan koneksi ke server PVE.
- Baris 11: digunakan agar tidak menampilkan hasil dari eksekusi *task* ketika playbook dijalankan menggunakan mode *verbose* atau *-v* sehingga menyembunyikan informasi sensitif yaitu *variable* terkait koneksi ke server PVE.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

```
12
13     - name: Membuat container
14       include_tasks: membuat-container.yml
15       loop: "{{ siswa }}"
16       when: aksi == "buat"
17
18     - name: Membuat user
19       include_tasks: membuat-user.yml
20       loop: "{{ siswa }}"
21       when: aksi == "buat"
22
23     - name: Membuat permission
24       include_tasks: membuat-permission.yml
25       loop: "{{ siswa }}"
26       when: aksi == "buat"
```

Baris 13: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 14: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**membuat-container.yml**” yang didalamnya memuat instruksi untuk membuat *container* baru.
- Baris 15: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Membuat container**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terbentuk *container* masing-masing dengan **vmid** 101, 102 dan 103 serta 104.
- Baris 16: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Membuat container**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**buat**”.
- Baris 18: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat user**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 19: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**membuat-user.yml**” yang didalamnya memuat instruksi untuk membuat *user* baru.
- Baris 20: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Membuat user**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terbentuk *user* masing-masing dengan *user name* 2501, 2502 dan 2503 serta 2504.
- Baris 21: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Membuat user**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**buat**”.
- Baris 23: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat permission**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 24: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**membuat-permission.yml**” yang didalamnya memuat instruksi untuk membuat *permission* atau ijin akses baru.
- Baris 25: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Membuat permission**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terbentuk *permission* atau ijin akses bagi *user name* 2501, 2502 dan 2503 serta 2504 terhadap *container* ID 101, 102, 103 dan 104 dengan role **PVEVMUser**.
- Baris 26: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Membuat permission**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**buat**”.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

```
27
28     - name: Menghapus permission
29       include_tasks: menghapus-permission.yml
30       loop: "{{ siswa }}"
31       when: aksi == "hapus"
32
33     - name: Menghapus user
34       include_tasks: menghapus-user.yml
35       loop: "{{ siswa }}"
36       when: aksi == "hapus"
37
38     - name: Menghapus container
39       include_tasks: menghapus-container.yml
40       loop: "{{ siswa }}"
41       when: aksi == "hapus"
```

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 28: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus permission**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 29: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**menghapus-permission.yml**” yang didalamnya memuat instruksi untuk menghapus *permission* atau ijin akses dari *user* terhadap objek *container* di server PVE.
- Baris 30: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Menghapus permission**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terhapus *permission* atau ijin akses dari *user name* 2501, 2502 dan 2503 serta 2504 terhadap *container ID* 101, 102, 103 dan 104 dengan role **PVEVMUser**.
- Baris 31: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Menghapus permission**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**hapus**”.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 33: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus user**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 34: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**menghapus-user.yml**” yang didalamnya memuat instruksi untuk menghapus *user*.
- Baris 35: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Menghapus user**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terhapus *user* masing-masing dengan *user name* 2501, 2502 dan 2503 serta 2504.
- Baris 36: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Menghapus user**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**hapus**”.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 38: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 39: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**menghapus-container.yml**” yang didalamnya memuat instruksi untuk menghapus *container*.
- Baris 40: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Menghapus container**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terhapus *container* masing-masing dengan **vmid** 101, 102 dan 103 serta 104.
- Baris 41: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Menghapus container**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**hapus**”.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

```
42
43 - name: Manajemen container
44   include_tasks: power-management-container.yml
45   loop: "{{ siswa }}"
46   when: aksi == "started" or aksi == "stopped" or aksi == "restarted"
47
```

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

- Baris 43: **name** digunakan untuk menentukan nama dari *task* yaitu “**Manajemen container container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 44: **include_tasks** digunakan untuk menyisipkan *file task* dengan nama “**power-manajemen-container.yml**” yang didalamnya memuat instruksi untuk memmanajemen *container*.

MEMBUAT FILE ANSIBLE PLAYBOOK BERNAMA PROXMOX-IAC.YML

- Baris 45: **loop** digunakan untuk melakukan perulangan eksekusi *task* “**Menghapus container**” sejumlah nilai yang terdapat pada *variable* “**siswa**” yaitu sebanyak 4 (empat) kali sehingga akan terhapus *container* masing-masing dengan **vmid** 101, 102 dan 103 serta 104.
- Baris 46: **when** merupakan pernyataan kondisional atau kendali program agar *task* “**Manajemen container**” ini hanya akan dijalankan ketika *variable* aksi bernilai “**started**” untuk menjalankan *container* atau “**restarted**” untuk menghentikan dan menjalankan kembali *container* (*reboot*) atau “**stopped**” untuk menghentikan *container*.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan tombol **Enter**.

Keluar dari editor *nano* dengan menekan tombol **CTRL+X**.

MEMBUAT FILE MEMBUAT-CONTAINER.YML

File **membuat-container.yml** memuat *task* yang digunakan untuk membuat *container*.

```
nano -l membuat-container.yml
```

```
GNU nano 7.2                          membuat-container.yml
1  name: Membuat Container
2  community.general.proxmox:
3    api_host: "{{ api_host }}"
4    api_user: "{{ api_user }}"
5    api_password: "{{ api_password }}"
6    node: "{{ node }}"
7    vmid: "{{ item.container.vmid }}"
8    hostname: "{{ item.container.hostname }}"
9    password: "{{ ct_root_password }}"
10   ostemplate: 'local:vztmpl/{{ ct_ostemplate }}'
11   cpus: 1
12   cpuunits: 500
13   memory: 512
14   swap: 512
15   disk: 'local:5'
16   nameserver: "{{ ct_nameserver }}"
17   searchdomain: "{{ ct_searchdomain }}"
18   netif: '{"net0": "name=eth0,gw={{ ct_gateway }},ip={{ item.container.ip_address }},bridge=vmb0"}'
19   onboot: no
20   unprivileged: yes
21   state: present
22
```

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

MEMBUAT FILE MEMBUAT-CONTAINER.YML

- Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat Container**”. Di dalam setiap *task* memuat kode yang berhubungan dengan modul yang harus dieksekusi.
- Baris 2: nama *module* yang dieksekusi yaitu **community.general.proxmox**.
- Baris 3: merupakan parameter dari **module proxmox** untuk menentukan *host* dari PVE yang nilainya diambil dari *variable* “{{ api_host }}”.
- Baris 4: merupakan parameter dari **module proxmox** untuk menentukan *user* yang digunakan login ke PVE dimana nilainya diambil dari *variable* “{{ api_user }}”.
- Baris 5: merupakan parameter dari **module proxmox** untuk menentukan password otentikasi ke PVE yang nilainya diambil dari *variable* “{{ api_password }}”.
- Baris 6: merupakan parameter dari **module proxmox** untuk menentukan *node* dari PVE yang nilainya diambil dari *variable* “{{ node }}”.
- Baris 7: digunakan untuk mengatur **ID instance** atau **container** yang nilainya diambil dari *variable* “{{ item.container.vmid }}”.

MEMBUAT FILE MEMBUAT-CONTAINER.YML

- Baris 8: digunakan untuk mengatur **hostname** dari *instance* atau *container* yang nilainya diambil dari *variable* "{{ item.container.hostname }}".
- Baris 9: digunakan untuk mengatur sandi untuk **user root** pada *container* yang nilainya diambil dari *variable* "{{ ct_root_password }}".
- Baris 10: digunakan untuk mengatur **container template** yang digunakan untuk pembuatan *container* dimana nilainya diambil dari *variable* "{{ ct_ostemplate }}".
- Baris 11: digunakan untuk mengatur jumlah CPU yang dialokasikan untuk *container* yaitu 1.
- Baris 12: digunakan untuk mengatur CPU weight dari *container*, sebagai contoh 500.
- Baris 13: digunakan untuk mengatur ukuran memori dari *container* dalam satuan *megabyte*, sebagai contoh 512.
- Baris 14: digunakan untuk mengatur ukuran dari **swap** dalam satuan *megabyte*, sebagai contoh 512.

MEMBUAT FILE MEMBUAT-CONTAINER.YML

- Baris 15: digunakan untuk mengatur media penyimpanan dari *container* yaitu sebagai contoh **local** dan juga mengatur ukuran dari *hardisk* dalam satuan *gigabyte*, sebagai contoh 5.
- Baris 16: digunakan untuk mengatur alamat IP dari *server* DNS yang nilainya diambil dari *variable* "{{ ct_name_server }}".
- Baris 17: digunakan untuk nama **domain** pencarian untuk *container* yang nilainya diambil dari *variable* "{{ ct_search_domain }}".
- Baris 18: digunakan untuk mengatur **network interface** meliputi nama dari **interface**, alamat **IP** yang nilainya diambil dari *variable* "{{ item.container.ip_address }}", **default gateway** yang nilainya diambil dari *variable* "{{ ct_gateway }}" dan **bridge**.
- Baris 19: digunakan untuk mengatur apakah *container* akan dijalankan (**started**) ketika *host* dari sistem melakukan **boot** yaitu berupa nilai *boolean* “yes” dan “no”. Sebagai contoh diatur **no**.

MEMBUAT FILE MEMBUAT-CONTAINER.YML

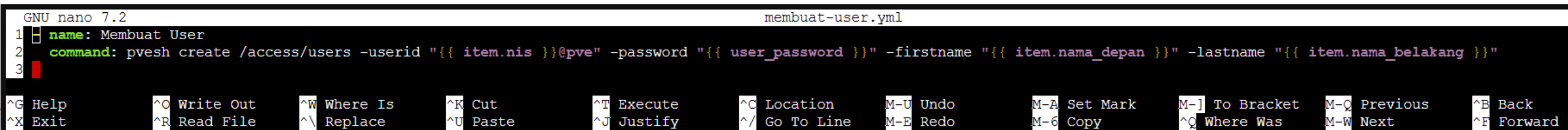
- Baris 20: digunakan untuk mengatur pemetaan semua **user ID (UID)** dan **group ID (GID)** pada *container* ke rentang nomor berbeda pada *host machine*. Umumnya **root** memiliki **uid 0** dipetakan menjadi **uid 100000**, **user id 1** menjadi **100001** dan seterusnya. Nilai dari parameter ini berupa *boolean* yaitu **yes** dan **no**. Sebagai contoh diatur **yes**.
- Baris 21: digunakan untuk mengatur **state** dari *container* yang meliputi **present** (membuat container), **started** (menjalankan), **absent** (menghapus container), **stopped** (menghentikan) dan **restarted** (menjalankan ulang atau reboot). Untuk membuat *container* maka **state** yang digunakan adalah **present**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE MEMBUAT-USER.YML

File **membuat-user.yml** memuat *task* yang digunakan untuk membuat *user*.

```
nano -l membuat-user.yml
```



```
GNU nano 7.2 membuat-user.yml
1 name: Membuat User
2 command: pvesh create /access/users -userid "{{ item.nis }}"@pve" -password "{{ user_password }}" -firstname "{{ item.nama_depan }}" -lastname "{{ item.nama_belakang }}"
3
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark  M-] To Bracket M-Q Previous  ^B Back
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/_ Go To Line  M-E Redo      M-6 Copy      ^Q Where Was  M-W Next     ^F Forward
```

Penjelasan:

- Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat user**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 2: nama *module* yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait pembuatan *user* baru di server PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE MEMBUAT-PERMISSION.YML

File **membuat-permission.yml** memuat *task* yang digunakan untuk mengatur ijin akses dari setiap user terhadap *object container* di server **PVE**.

```
nano -l membuat-permission.yml
```

```
GNU nano 7.2                          membuat-permission.yml
1  name: Membuat Permission
2  command: pvesh set /access/acl -path "/vms/{{ item.container.vmid }}" -users "{{ item.nis }}@pve" -roles "{{ proles }}"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste       ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Penjelasan:

- Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Membuat Permission**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 2: nama *module* yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait pembuatan **permissions** atau ijin akses dari setiap user terhadap objek tertentu di server PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE MENGHAPUS-PERMISSION.YML

File **menghapus-permission.yml** memuat *task* yang digunakan untuk menghapus ijin akses dari setiap *user* terhadap *object* di **server PVE**.

```
nano -l menghapus-permission.yml
```

```
GNU nano 7.2                                     menghapus-permission.yml
1  name: Menghapus Permission
2  command: pvesh set /access/acl -delete -path "/vms/{{ item.container.vmid }}" -users "{{ item.nis }}@pve" -roles "{{ proles }}"

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo      M-A Set Mark
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line M-E Redo      M-6 Copy
```

Penjelasan:

- Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus Permission**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 2: nama *module* yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait penghapusan **permissions** atau ijin akses dari setiap user terhadap objek tertentu di server PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE MENGHAPUS-USER.YML

File **menghapus-user.yml** memuat *task* yang digunakan untuk menghapus data *user*.

```
nano -l menghapus-user.yml
```

```
GNU nano 7.2                                     menghapus-user.yml
1  name: Menghapus User
2  command: pvesh delete /access/users/"{{ item.nis }}@pve"

^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
```

Penjelasan:

- Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus user**”. Di dalam setiap *task* memuat kode yang berhubungan dengan *module* yang harus dieksekusi.
- Baris 2: nama *module* yang dieksekusi yaitu **command**. Modul ini digunakan untuk mengeksekusi utilitas **pvesh** terkait penghapusan *user* di server PVE.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE MENGHAPUS-CONTAINER.YML

File **menghapus-container.yml** memuat *task* yang digunakan untuk menghapus *container*.

```
nano -l menghapus-container.yml
```

```
GNU nano 7.2                                     menghapus-container.yml
1  name: Menghapus container
2  community.general.proxmox:
3    api_host: "{{ api_host }}"
4    api_user: "{{ api_user }}"
5    api_password: "{{ api_password }}"
6    node: "{{ node }}"
7    vmid: "{{ item.container.vmid }}"
8    state: absent
9
^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
```

Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Menghapus container**”.
Di dalam setiap *task* memuat kode yang berhubungan dengan modul yang harus dieksekusi.

MEMBUAT FILE MENGHAPUS-CONTAINER.YML

- Baris 2: nama *module* yang dieksekusi yaitu **community.general.proxmox**.
- Baris 3: merupakan parameter dari **module proxmox** untuk menentukan *host* dari PVE yang nilainya diambil dari *variable* “{{ api_host }}”.
- Baris 4: merupakan parameter dari **module proxmox** untuk menentukan *user* yang digunakan login ke PVE dimana nilainya diambil dari *variable* “{{ api_user }}”.
- Baris 5: merupakan parameter dari **module proxmox** untuk menentukan password otentikasi ke PVE yang nilainya diambil dari *variable* “{{ api_password }}”.
- Baris 6: merupakan parameter dari **module proxmox** untuk menentukan *node* dari PVE yang nilainya diambil dari *variable* “{{ node }}”.
- Baris 7: digunakan untuk mengatur **ID instance** atau **container** yang akan dihapus dimana nilainya diambil dari *variable* “{{ item.container.vmid }}”.

MEMBUAT FILE MENGHAPUS-CONTAINER.YML

- Baris 8: digunakan untuk menghapus *container* dengan mengatur nilai dari parameter **state** menjadi **absent**.

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

MEMBUAT FILE POWER-MANAGEMENT-CONTAINER.YML

File **power-management-container.yml** memuat **task** yang digunakan untuk menjalankan (**started**) dan menjalankan ulang (**restarted**) serta menghentikan (**stopped**).

```
nano -l power-management-container.yml
```

```
GNU nano 7.2                                power-management-container.yml
1  name: Power management container
2  community.general.proxmox:
3    api_host: "{{ api_host }}"
4    api_user: "{{ api_user }}"
5    api_password: "{{ api_password }}"
6    node: "{{ node }}"
7    vmid: "{{ item.container.vmid }}"
8    state: "{{ aksi }}"
9
^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
```

Baris 1: **name** digunakan untuk menentukan nama dari *task* yaitu “**Power management container**”.
Di dalam setiap *task* memuat kode yang berhubungan dengan modul yang harus dieksekusi.

MEMBUAT FILE POWER-MANAGEMENT-CONTAINER.YML

- Baris 2: nama *module* yang dieksekusi yaitu **community.general.proxmox**.
- Baris 3: merupakan parameter dari **module proxmox** untuk menentukan *host* dari PVE yang nilainya diambil dari *variable* “{{ api_host }}”.
- Baris 4: merupakan parameter dari **module proxmox** untuk menentukan *user* yang digunakan login ke PVE dimana nilainya diambil dari *variable* “{{ api_user }}”.
- Baris 5: merupakan parameter dari **module proxmox** untuk menentukan password otentikasi ke PVE yang nilainya diambil dari *variable* “{{ api_password }}”.
- Baris 6: merupakan parameter dari **module proxmox** untuk menentukan *node* dari PVE yang nilainya diambil dari *variable* “{{ node }}”.
- Baris 7: digunakan untuk mengatur **ID instance** atau **container** yang akan dimanajemen dimana nilainya diambil dari *variable* “{{ item.container.vmid }}”.

MEMBUAT FILE POWER-MANAGEMENT-CONTAINER.YML

- Baris 8: digunakan untuk mengatur nilai dari parameter **state** yang diambil dari *variable* “`{{ aksi }}`”. Nilai dari *variable* **aksi** yang valid meliputi **started** (untuk menjalankan *container*), **stopped** (untuk menghentikan *container*) dan **restarted** (menjalankan ulang atau *container*).

Simpan perubahan dengan menekan tombol **CTRL+O** dan tekan **Enter**. Tekan tombol **CTRL+X** untuk keluar dari editor **nano**.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

`ansible-playbook proxmox-iac.yml`

Cuplikan hasil eksekusi *playbook* adalah sebagai berikut:

```
root@ansible:~/proxmox-iac# ansible-playbook proxmox-iac.yml

PLAY [Proxmox IaC] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [include vars] *****
ok: [pve]
```

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

```
TASK [Membuat container] *****
included: /root/proxmox-iac/membuat-container.yml for pve => (item={'nis': 2501, 'nama_d
eplan': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'serve
r101.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/membuat-container.yml for pve => (item={'nis': 2502, 'nama_d
eplan': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': '
server102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/membuat-container.yml for pve => (item={'nis': 2503, 'nama_d
eplan': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'serve
r103.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/membuat-container.yml for pve => (item={'nis': 2504, 'nama_d
eplan': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104
.belajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Membuat Container] *****
changed: [pve]

TASK [Membuat Container] *****
changed: [pve]

TASK [Membuat Container] *****
changed: [pve]

TASK [Membuat Container] *****
changed: [pve]
```

Terlihat *task* **Membuat Container** memiliki **state *changed*** yang bermakna terjadi perubahan pada server **PVE** yaitu pembuatan *container*.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

```
TASK [Membuat user] *****
included: /root/proxmox-iac/membuat-user.yml for pve => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/membuat-user.yml for pve => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/membuat-user.yml for pve => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/membuat-user.yml for pve => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Membuat User] *****
changed: [pve]

TASK [Membuat User] *****
changed: [pve]

TASK [Membuat User] *****
changed: [pve]

TASK [Membuat User] *****
changed: [pve]
```

Terlihat *task* **Membuat User** memiliki **state** **changed** yang bermakna terjadi perubahan pada server **PVE** yaitu pembuatan *user*.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

```
TASK [Membuat permission] *****
included: /root/proxmox-iac/membuat-permission.yml for pve => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/membuat-permission.yml for pve => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/membuat-permission.yml for pve => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/membuat-permission.yml for pve => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Membuat Permission] *****
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]

TASK [Membuat Permission] *****
changed: [pve]
```

Terlihat *task* **Membuat Permission** memiliki **state *changed*** yang bermakna terjadi perubahan pada server **PVE** yaitu pembuatan ijin akses untuk setiap *user* terhadap objek tertentu.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

```
TASK [Menghapus permission] *****
skipping: [pvel] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
skipping: [pvel] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pvel] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
skipping: [pvel] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pvel]

TASK [Menghapus user] *****
skipping: [pvel] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
skipping: [pvel] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pvel] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
skipping: [pvel] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pvel]
```

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK PEMBUATAN CONTAINERS, USERS DAN PERMISSIONS

```
TASK [Menghapus container] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]

TASK [Manajemen container] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]

PLAY RECAP *****
pve                : ok=26   changed=12   unreachable=0   failed=0   skipped=4
                    rescued=0   ignored=0
```

MEMVERIFIKASI HASIL PEMBUATAN CONTAINERS

```
ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

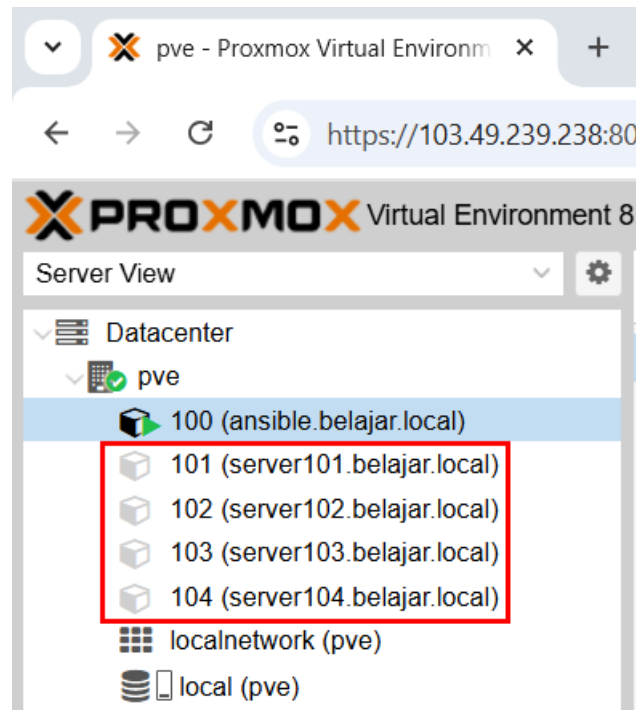
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

status	vmid	cpu	cpus	disk	diskread	diskwrite	lock	maxdisk	maxmem	maxswap	mem	name	netin	netout	tags	template	uptime
running	100	0	1	1.41 GiB	364.33 MiB	9.00 MiB		3.86 GiB	1.00 GiB	1.00 GiB	121.52 MiB	ansible.belajar.local	111.38 KiB	1.59 MiB			1h 38m 31s
stopped	104	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server104.belajar.local	0.00 B	0.00 B			0s
stopped	102	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server102.belajar.local	0.00 B	0.00 B			0s
stopped	103	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server103.belajar.local	0.00 B	0.00 B			0s
stopped	101	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server101.belajar.local	0.00 B	0.00 B			0s

Terlihat **CT ID 101-104** telah berhasil dibuat.

MEMVERIFIKASI HASIL PEMBUATAN CONTAINERS

Pada panel sebelah kiri dari **Web GUI PVE** terlihat **CT ID 101-104** telah berhasil dibuat.



MEMVERIFIKASI HASIL PEMBUATAN USERS

ansible servers -m command -a "pvesh get /access/users"

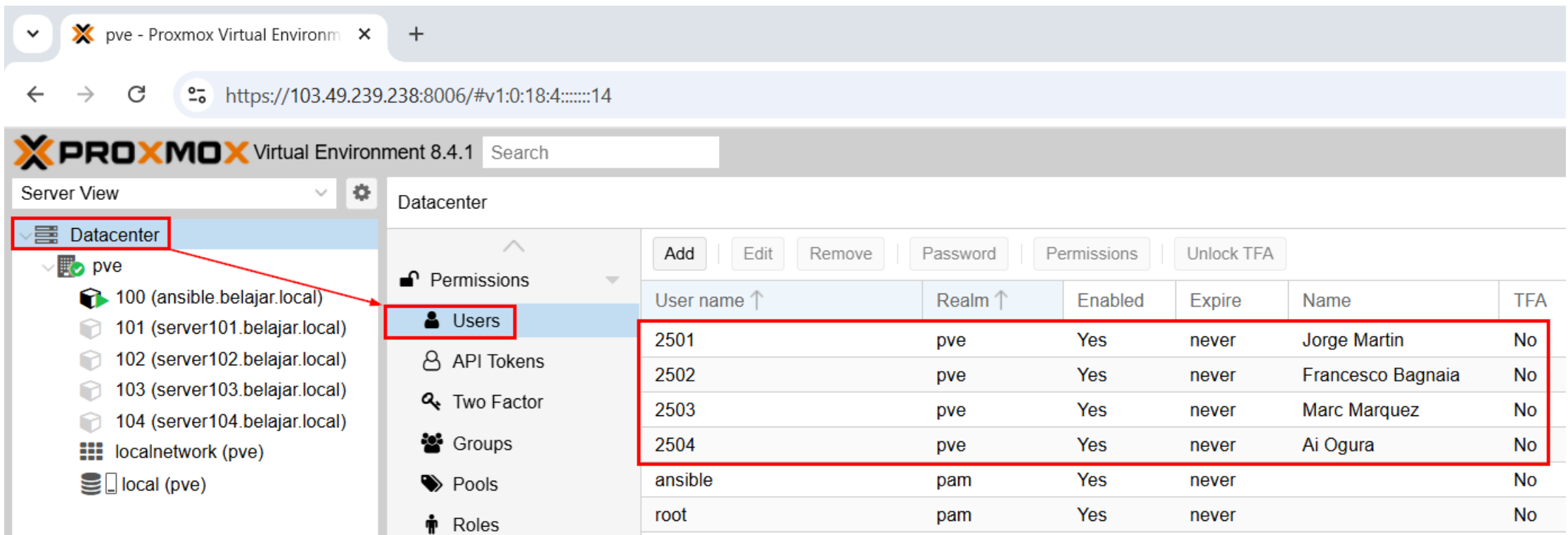
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /access/users"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

userid	comment	email	enable	expire	firstname	groups	keys	lastname	realm-type	tfa-locked-until	tokens	totp-locked
2501@pve			1	0	Jorge			Martin	pve			
2502@pve			1	0	Francesco			Bagnaia	pve			
2503@pve			1	0	Marc			Marquez	pve			
2504@pve			1	0	Ai			Ogura	pve			
ansible@pam			1	0					pam			
root@pam			1	0					pam			

Terlihat user dengan **user name 2501-2504** telah berhasil dibuat.

MEMVERIFIKASI HASIL PEMBUATAN USERS

Pada panel sebelah kiri dari **Web GUI PVE**, pilih **Datacenter** kemudian pada panel detail sebelah kanan pilih **Permissions** → **Users**. Terlihat *user* dengan **user name** **2501-2504** telah berhasil dibuat.



The screenshot shows the Proxmox Web GUI interface. On the left sidebar, the 'Datacenter' menu item is highlighted with a red box. A red arrow points from this menu item to the 'Users' menu item in the 'Datacenter' panel, which is also highlighted with a red box. The main panel displays a table of users. The table has columns for 'User name', 'Realm', 'Enabled', 'Expire', 'Name', and 'TFA'. The users listed are 2501, 2502, 2503, and 2504, all with the realm 'pve' and 'Enabled' status 'Yes'. The names are Jorge Martin, Francesco Bagnaia, Marc Marquez, and Ai Ogura respectively. The 'TFA' column shows 'No' for all users. The table is also highlighted with a red box.

User name ↑	Realm ↑	Enabled	Expire	Name	TFA
2501	pve	Yes	never	Jorge Martin	No
2502	pve	Yes	never	Francesco Bagnaia	No
2503	pve	Yes	never	Marc Marquez	No
2504	pve	Yes	never	Ai Ogura	No
ansible	pam	Yes	never		No
root	pam	Yes	never		No

MEMVERIFIKASI HASIL PEMBUATAN CONTAINERS

```
ansible servers -m command -a "pvesh get /access/acl"
```

```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /access/acl"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

path	roleid	type	ugid	propagate
/	Administrator	user	ansible@pam	1
/vms/101	PVEVMUser	user	2501@pve	1
/vms/102	PVEVMUser	user	2502@pve	1
/vms/103	PVEVMUser	user	2503@pve	1
/vms/104	PVEVMUser	user	2504@pve	1

Terlihat ijin akses untuk setiap *user* terhadap setiap objek yang ditetapkan telah berhasil dibuat.

MEMVERIFIKASI HASIL PEMBUATAN PERMISSIONS

Pada panel sebelah kiri dari **Web GUI PVE**, pilih **Datacenter** kemudian pada panel detail sebelah kanan pilih **Permissions**.

PROXMOX Virtual Environment 8.4.1

Server View

Datacenter

Replication

Permissions

Users

API Tokens

Two Factor

Groups

Path ↑	User/Group/API Token	Role	Propagate
/	ansible@pam	Administrator	true
/vms/101	2501@pve	PVEVMUser	true
/vms/102	2502@pve	PVEVMUser	true
/vms/103	2503@pve	PVEVMUser	true
/vms/104	2504@pve	PVEVMUser	true

Terlihat ijin akses untuk setiap *user* terhadap setiap objek yang ditetapkan telah berhasil dibuat.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENJALANKAN CONTAINERS

```
ansible-playbook proxmox-iac.yml -e aksi=started
```

Option atau parameter `-e` digunakan untuk mengatur *environment variable* yaitu nilai dari *variable* **aksi** menjadi **started**.

Cuplikan hasil eksekusi *playbook* adalah sebagai berikut:

```
root@ansible:~/proxmox-iac# ansible-playbook proxmox-iac.yml -e aksi=started

PLAY [Proxmox IaC] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [include vars] *****
ok: [pve]
```

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENJALANKAN CONTAINERS

```
TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=10   changed=4   unreachable=0   failed=0   skipped=6
                    rescued=0   ignored=0
```

Terlihat *task* **Power management container** memiliki **state** **changed** yang bermakna terjadi perubahan pada server **PVE** yaitu *container* dijalankan (**started**).

MEMVERIFIKASI HASIL MENJALANKAN SELURUH CONTAINER

```
ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

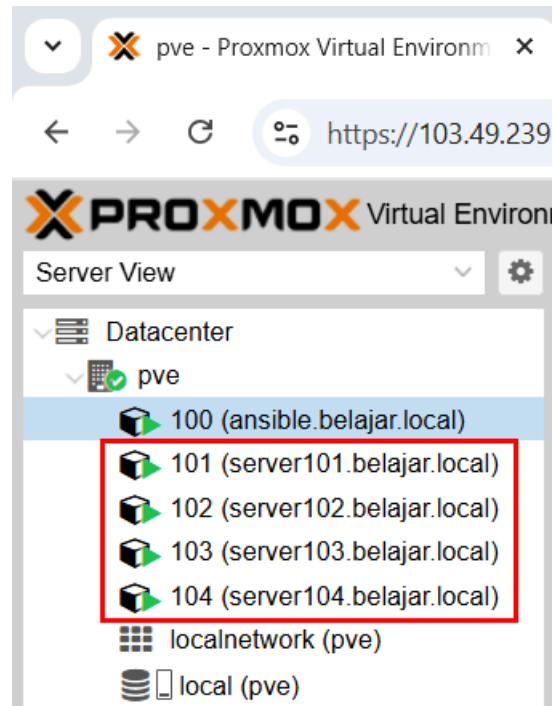
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

status	vmid	cpu	cpus	disk	diskread	diskwrite	lock	maxdisk	maxmem	maxswap	mem	name	netin	netout	tags	template	uptime
running	102	0	2	582.21 MiB	190.93 MiB	10.84 MiB		4.84 GiB	512.00 MiB	512.00 MiB	33.83 MiB	server102.belajar.local	892.00 B	446.00 B			7m 49s
running	104	0	2	582.21 MiB	189.54 MiB	12.30 MiB		4.84 GiB	512.00 MiB	512.00 MiB	33.84 MiB	server104.belajar.local	0.00 B	446.00 B			7m 33s
running	100	0	1	1.41 GiB	358.43 MiB	5.12 MiB		3.86 GiB	1.00 GiB	1.00 GiB	116.27 MiB	ansible.belajar.local	58.21 KiB	818.50 KiB			17m 17s
running	101	0	2	582.20 MiB	191.28 MiB	10.55 MiB		4.84 GiB	512.00 MiB	512.00 MiB	33.77 MiB	server101.belajar.local	1.31 KiB	446.00 B			7m 55s
running	103	0	2	582.20 MiB	190.05 MiB	11.12 MiB		4.84 GiB	512.00 MiB	512.00 MiB	33.81 MiB	server103.belajar.local	446.00 B	446.00 B			7m 40s

Terlihat **CT ID 101-104** telah berhasil dijalankan.

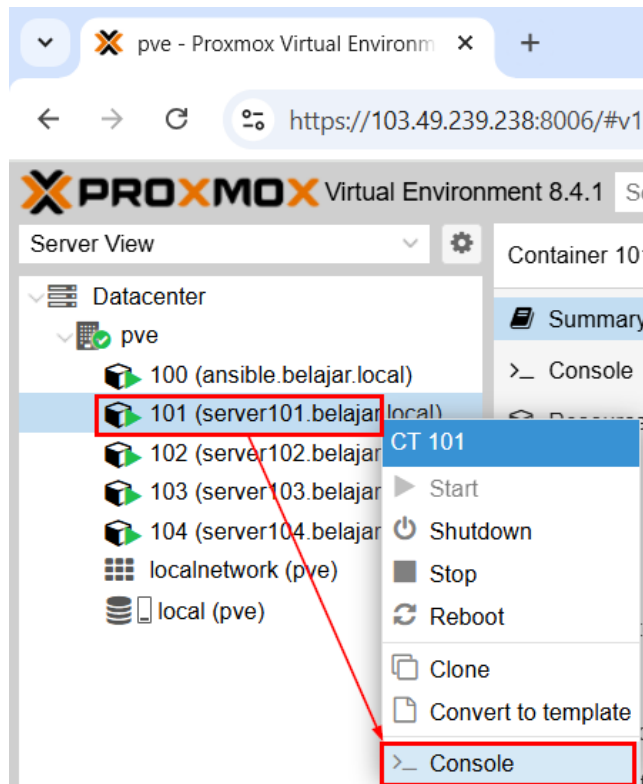
MEMVERIFIKASI HASIL MENJALANKAN SELURUH CONTAINER

Pada panel sebelah kiri dari **Web GUI PVE** terlihat **CT ID 101-104** telah berhasil dijalankan.

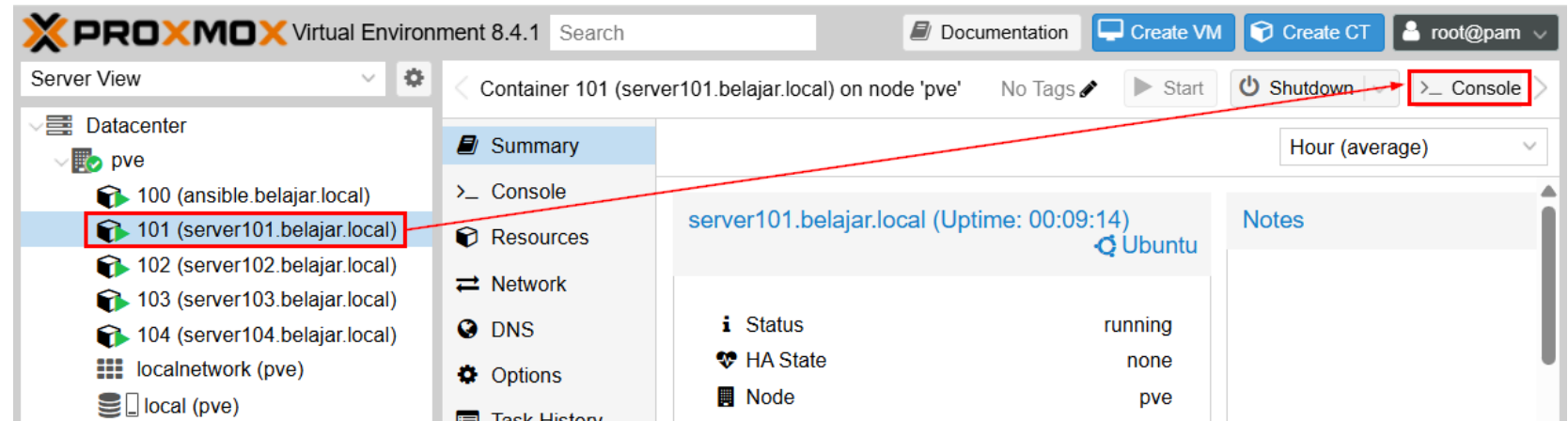


MEMVERIFIKASI AKSES CONSOLE KE SALAH SATU CONTAINER

Pada panel sebelah kiri dari *Web GUI PVE*, pilih **CT “101 (server101.belajar.local)”** yang terdapat dibawah *node pve* dari menu **Datacenter**. Selanjutnya lakukan klik kanan pada “101 (server101.belajar.local)” dan pilih **Console**.

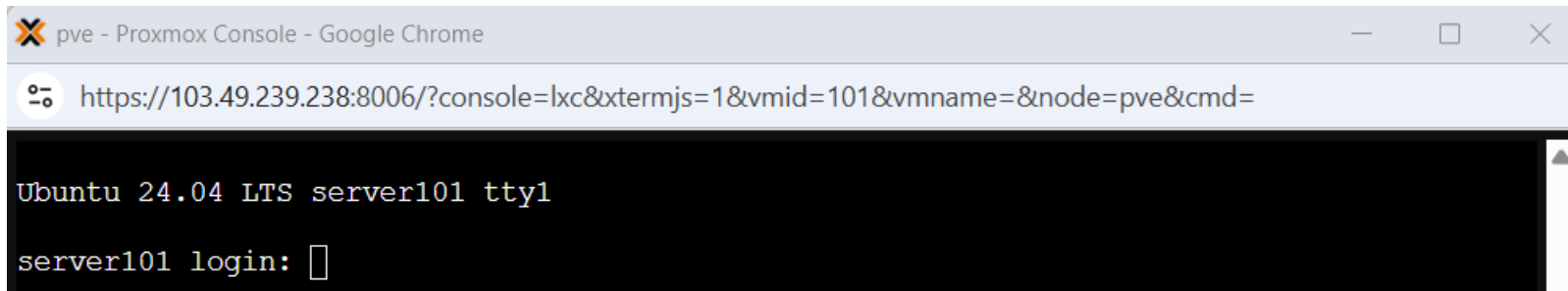


Atau dapat pula dengan cara memilih “101 (server101.belajar.local)” yang terdapat dibawah *node pve* dari menu **Datacenter** dan pada panel detail sebelah kanan menekan tombol **Console**.



MEMVERIFIKASI AKSES CONSOLE KE SALAH SATU CONTAINER

Tampil inputan **server101 login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari CT **Ubuntu 24.04**.

A screenshot of a web browser window titled "pve - Proxmox Console - Google Chrome". The address bar shows the URL "https://103.49.239.238:8006/?console=lxc&xtermjs=1&vmid=101&vmname=&node=pve&cmd=". The main content area is a black terminal window with white text. It displays "Ubuntu 24.04 LTS server101 tty1" on the first line and "server101 login: " followed by a cursor on the second line.

```
pve - Proxmox Console - Google Chrome
https://103.49.239.238:8006/?console=lxc&xtermjs=1&vmid=101&vmname=&node=pve&cmd=
Ubuntu 24.04 LTS server101 tty1
server101 login: 
```

Masukkan nama login “**root**” pada inputan **server101 Login** dan tekan tombol **Enter**. Tampil inputan **Password:**, masukkan sandi login dari user “**root**” yaitu “**mandalika**”, dan tekan tombol **Enter**.

MEMVERIFIKASI AKSES CONSOLE KE SALAH SATU CONTAINER

Apabila proses otentikasi login berhasil dilakukan maka akan tampil **prompt CLI** dari **CT server101** yang ditandai dengan tanda **#**.

```
Ubuntu 24.04 LTS server101 tty1

server101 login: root
Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.12-10-pve x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

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

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@server101:~#
```

MEMVERIFIKASI AKSES CONSOLE KE SALAH SATU CONTAINER

Verifikasi akses Internet dari container dilakukan ke salah satu domain, sebagai contoh **google.com**.

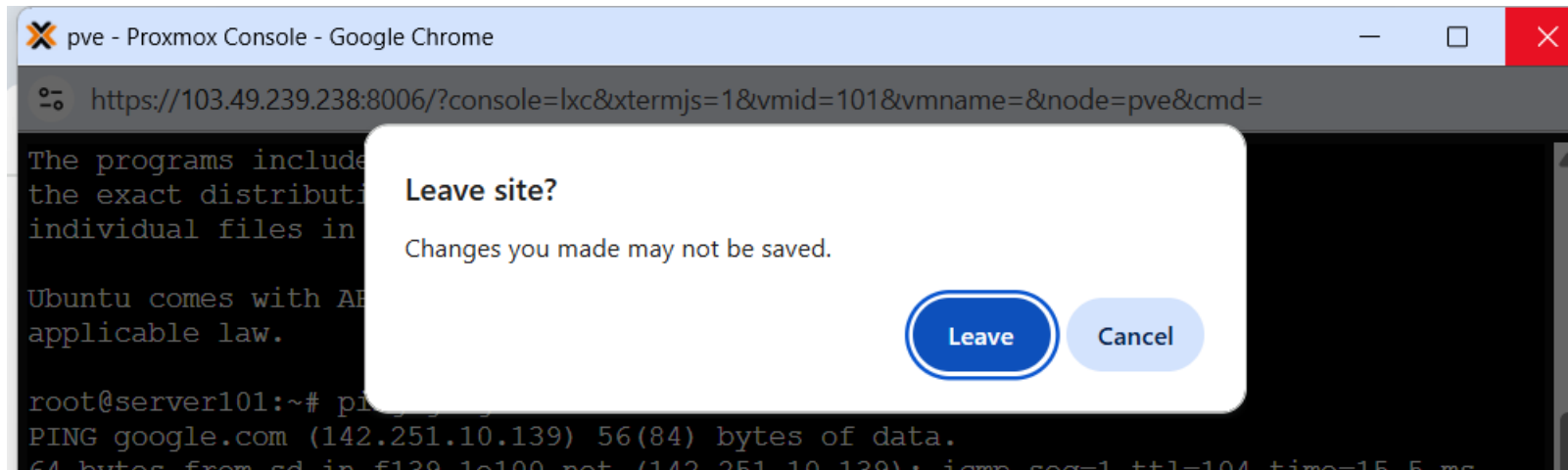
```
root@server101:~# ping google.com -c 4
PING google.com (142.251.10.139) 56(84) bytes of data.
64 bytes from sd-in-f139.1e100.net (142.251.10.139): icmp_seq=1 ttl=104 time=15.5 ms
64 bytes from sd-in-f139.1e100.net (142.251.10.139): icmp_seq=2 ttl=104 time=14.8 ms
64 bytes from sd-in-f139.1e100.net (142.251.10.139): icmp_seq=3 ttl=104 time=14.9 ms
64 bytes from sd-in-f139.1e100.net (142.251.10.139): icmp_seq=4 ttl=104 time=14.7 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 14.743/14.974/15.508/0.310 ms
root@server101:~#
```

Terlihat koneksi ke **google.com** berhasil dilakukan.

MEMVERIFIKASI AKSES CONSOLE KE SALAH SATU CONTAINER

- Keluar dari akses *container* dengan mengeksekusi perintah:
`exit`
- Tutup kotak dialog **Console** dari *container*. Tampil kotak dialog konfirmasi **Leave site?**. Tekan tombol **Leave**.



UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHENTIKAN CONTAINERS

```
ansible-playbook proxmox-iac.yml -e aksi=stopped
```

Option atau parameter `-e` digunakan untuk mengatur *environment variable* yaitu nilai dari *variable* **aksi** menjadi **stopped**.

Cuplikan hasil eksekusi *playbook* adalah sebagai berikut:

```
root@ansible:~/proxmox-iac# ansible-playbook proxmox-iac.yml -e aksi=stopped

PLAY [Proxmox IaC] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [include vars] *****
ok: [pve]
```

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHENTIKAN CONTAINERS

```
TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

TASK [Power management container] *****
changed: [pve]

PLAY RECAP *****
pve                : ok=10   changed=4    unreachable=0    failed=0    skipped=6
                   rescued=0    ignored=0
```

Terlihat *task* **Power management container** memiliki **state** **changed** yang bermakna terjadi perubahan pada server **PVE** yaitu *container* dihentikan (**stopped**).

MEMVERIFIKASI HASIL PENGHENTIAN SELURUH CONTAINER

```
ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

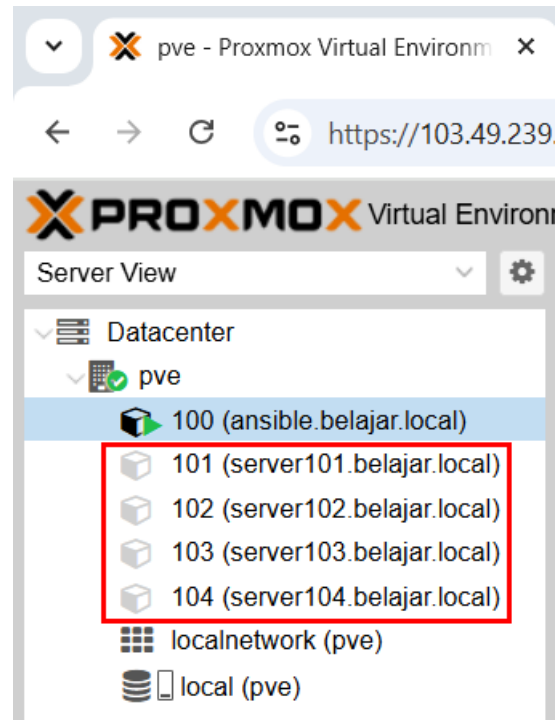
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could change the meaning of that path. See
https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

status	vmid	cpu	cpus	disk	diskread	diskwrite	lock	maxdisk	maxmem	maxswap	mem	name	netin	netout	tags	template	uptime
running	100	0	1	1.41 GiB	693.29 MiB	292.93 MiB		3.86 GiB	1.00 GiB	1.00 GiB	130.62 MiB	ansible.belajar.local	5.34 MiB	3.54 MiB			2h 57m 40s
stopped	101	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server101.belajar.local	0.00 B	0.00 B			0s
stopped	102	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server102.belajar.local	0.00 B	0.00 B			0s
stopped	104	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server104.belajar.local	0.00 B	0.00 B			0s
stopped	103	0	2	0.00 B	0.00 B	0.00 B		5.00 GiB	512.00 MiB	512.00 MiB	0.00 B	server103.belajar.local	0.00 B	0.00 B			0s

Terlihat **CT ID 101-104** telah berhasil dihentikan.

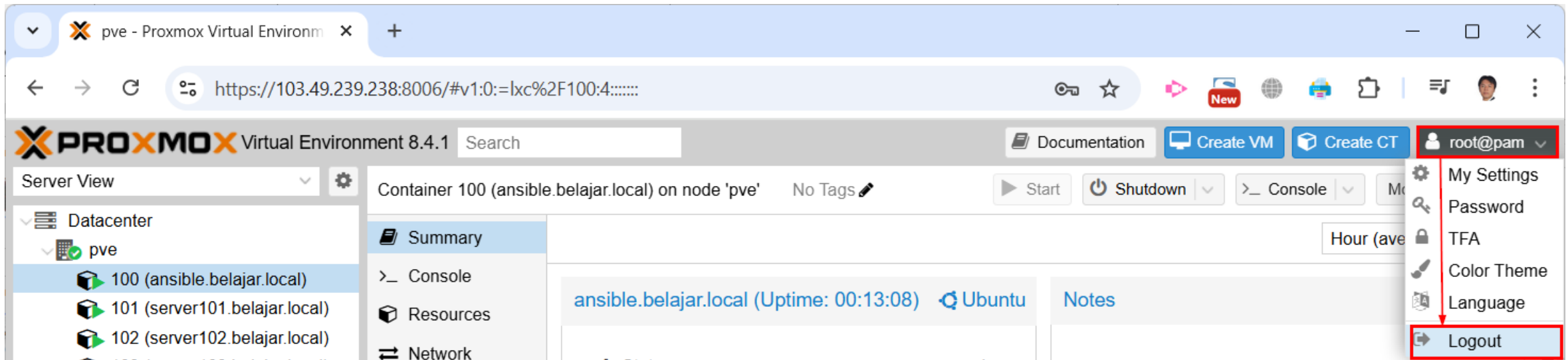
MEMVERIFIKASI HASIL PENGHENTIAN SELURUH CONTAINER

Pada panel sebelah kiri dari **Web GUI PVE** terlihat **CT ID 101-104** telah berhasil dihentikan.



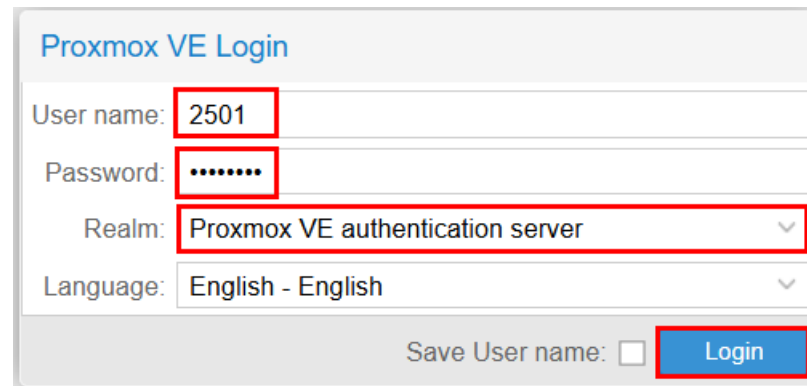
MEMVERIFIKASI HASIL PENGATURAN PERMISSION BAGI USER BARU PADA PVE

Keluar dari *Web GUI Proxmox* dengan memilih dropdown menu **root@pam** yang terdapat pada pojok kanan atas dan pilih **Logout**.

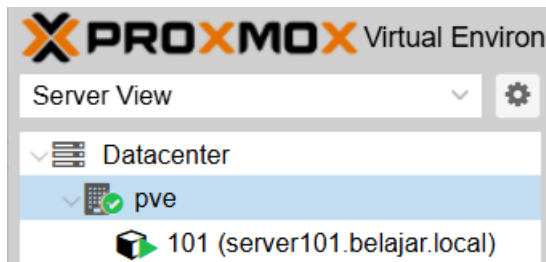


MEMVERIFIKASI HASIL PENGATURAN PERMISSION BAGI USER BARU PADA PVE

Lakukan login kembali ke *PVE WebGUI* menggunakan user “**2501**” dengan password “**bumigora**” dan realm “**Proxmox VE Authentication Server**”.

A screenshot of the Proxmox VE Login form. The form has a title 'Proxmox VE Login' in blue. It contains four input fields: 'User name:' with the value '2501', 'Password:' with masked characters '.....', 'Realm:' with a dropdown menu showing 'Proxmox VE authentication server', and 'Language:' with a dropdown menu showing 'English - English'. At the bottom, there is a checkbox for 'Save User name:' and a blue 'Login' button. Red rectangular boxes highlight the 'User name', 'Password', 'Realm', and 'Login' button fields.

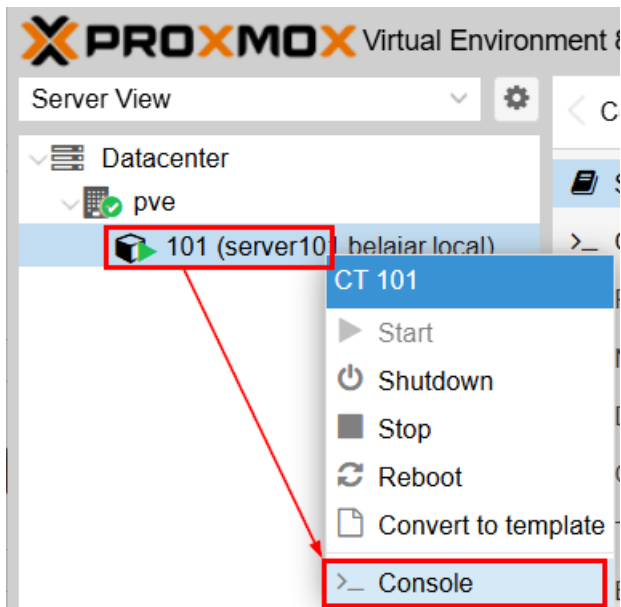
Klik tombol **Login**. Apabila login sukses maka akan tampil *Dashboard PVE*. Pada panel sebelah kiri pilih **Datacenter** > **pve** maka akan terlihat **CT ID 101**.



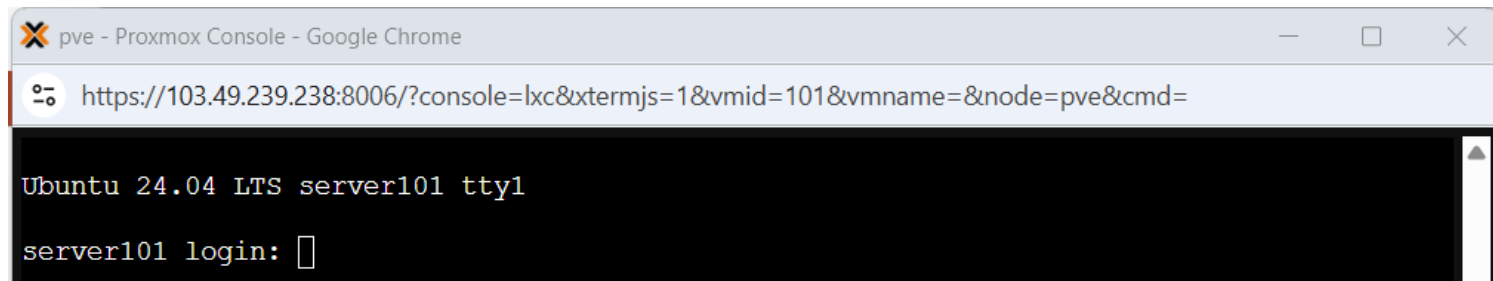
Hal ini sesuai dengan ijin akses yang diberikan pada user “**2501**” yaitu hanya dapat mengakses objek **CT ID 101 (server101.belajar.local)**.

MEMVERIFIKASI HASIL PENGATURAN PERMISSION BAGI USER BARU PADA PVE

Apabila CT telah berjalan maka lakukan akses **Console** dengan cara klik kanan pada “**101 (server101.belajar.local)**” yang terdapat di bawah *node* “**pve**” dari menu **Datacenter** dan pilih **Console**.



Tampil inputan **server101 login** untuk proses otentikasi sebelum pengguna dapat mengakses **Command Line Interface (CLI)** dari *server101*.

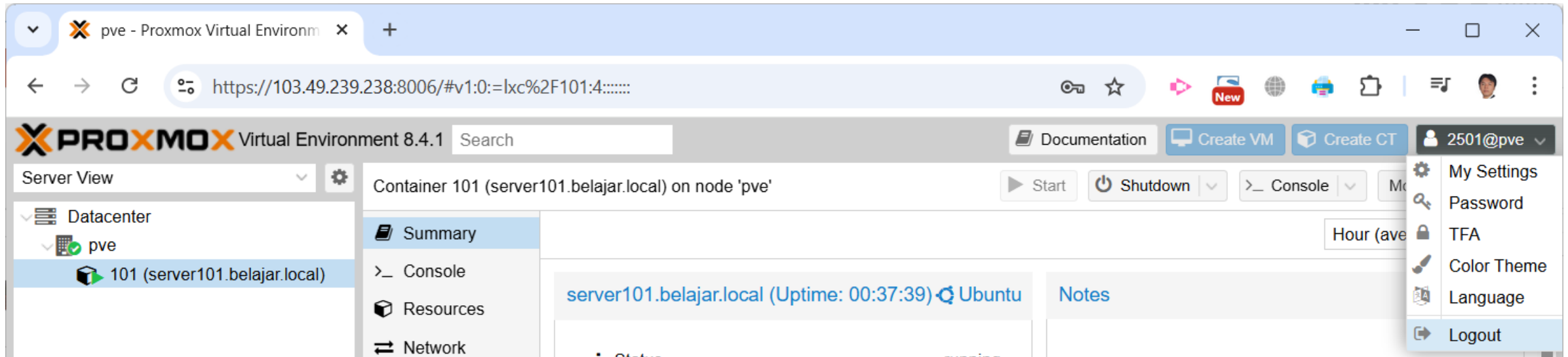


Tutup kotak dialog **Console**.

Sebaliknya jika **CT** masih berhenti maka jalankan terlebih dahulu dengan cara klik kanan pada “**101 (ansible.belajar.local)**” di bawah *node* “**pve**” dari menu **Datacenter** dan pilih **Start**. Baru lakukan akses **Console**.

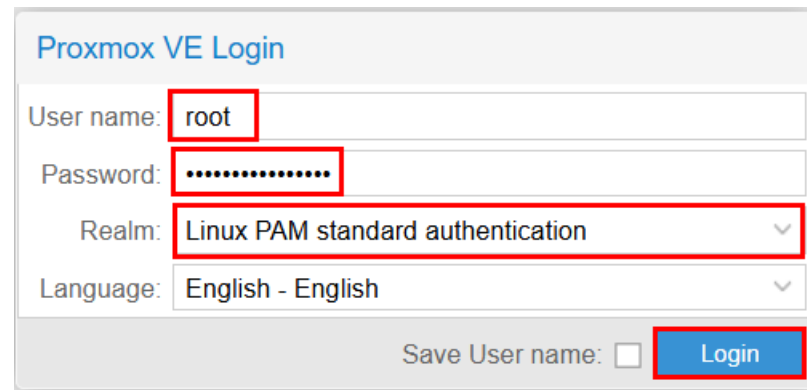
MEMVERIFIKASI HASIL PENGATURAN PERMISSION BAGI USER BARU PADA PVE

Keluar dari *Web GUI Proxmox* dengan memilih dropdown menu **2501@pve** yang terdapat pada pojok kanan atas dan pilih **Logout**.



MEMVERIFIKASI HASIL PENGATURAN PERMISSION BAGI USER BARU PADA PVE

Lakukan *login* kembali menggunakan user **root**. Pastikan jenis otentikasi (**Realm**) yang dipilih adalah **Linux PAM standard Authentication**.



Proxmox VE Login

User name: root

Password:

Realm: Linux PAM standard authentication

Language: English - English

Save User name: ☐ Login

Akses kembali **Console** dari **CT 100 (ansible.belajar.local)**.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
ansible-playbook proxmox-iac.yml -e aksi=hapus
```

Cuplikan hasil eksekusi *playbook* adalah sebagai berikut:

```
root@ansible:~/proxmox-iac# ansible-playbook proxmox-iac.yml -e aksi=hapus

PLAY [Proxmox IaC] *****

TASK [Gathering Facts] *****
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [pve]

TASK [include vars] *****
ok: [pve]

TASK [Membuat container] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.16
8.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.16
8.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]
```

Option atau parameter `-e` digunakan untuk mengatur *environment variable* yaitu nilai dari *variable* **aksi** menjadi **hapus**.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
TASK [Membuat user] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.16
8.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.16
8.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]

TASK [Membuat permission] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.16
8.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.16
8.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]
```

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
TASK [Menghapus permission] *****
included: /root/proxmox-iac/menghapus-permission.yml for pve => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/menghapus-permission.yml for pve => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/menghapus-permission.yml for pve => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/menghapus-permission.yml for pve => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Menghapus Permission] *****
changed: [pve]

TASK [Menghapus Permission] *****
changed: [pve]

TASK [Menghapus Permission] *****
changed: [pve]

TASK [Menghapus Permission] *****
changed: [pve]
```

Terlihat *task Menghapus Permission* memiliki **state *changed*** yang bermakna terjadi perubahan pada server **PVE** yaitu penghapusan ijin akses dari setiap user terhadap objek tertentu.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
TASK [Menghapus user] *****
included: /root/proxmox-iac/menghapus-user.yml for pve => (item={'nis': 2501, 'nama_depa
n': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'server10
1.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/menghapus-user.yml for pve => (item={'nis': 2502, 'nama_depa
n': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': 'ser
ver102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/menghapus-user.yml for pve => (item={'nis': 2503, 'nama_depa
n': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'server10
3.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/menghapus-user.yml for pve => (item={'nis': 2504, 'nama_depa
n': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104.be
lajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Menghapus User] *****
changed: [pve]

TASK [Menghapus User] *****
changed: [pve]

TASK [Menghapus User] *****
changed: [pve]

TASK [Menghapus User] *****
changed: [pve]
```

Terlihat *task*
Menghapus User
memiliki **state**
changed yang
bermakna terjadi
perubahan pada
server **PVE** yaitu
penghapusan *user*.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
TASK [Menghapus container] *****
included: /root/proxmox-iac/menghapus-container.yml for pve => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin', 'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
included: /root/proxmox-iac/menghapus-container.yml for pve => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagnaia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '192.168.0.102/24'}})
included: /root/proxmox-iac/menghapus-container.yml for pve => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez', 'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
included: /root/proxmox-iac/menghapus-container.yml for pve => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'container': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.104/24'}})

TASK [Menghapus container] *****
changed: [pve]

TASK [Menghapus container] *****
changed: [pve]

TASK [Menghapus container] *****
changed: [pve]

TASK [Menghapus container] *****
changed: [pve]
```

Terlihat *task Menghapus container* memiliki **state *changed*** yang bermakna terjadi perubahan pada server **PVE** yaitu penghapusan *container*.

UJICoba EKSEKUSI ANSIBLE PLAYBOOK UNTUK MENGHAPUS PERMISSIONS, USERS DAN CONTAINERS

```
TASK [Manajemen container] *****
skipping: [pve] => (item={'nis': 2501, 'nama_depan': 'Jorge', 'nama_belakang': 'Martin',
  'container': {'vmid': 101, 'hostname': 'server101.belajar.local', 'ip_address': '192.168.0.101/24'}})
skipping: [pve] => (item={'nis': 2502, 'nama_depan': 'Francesco', 'nama_belakang': 'Bagn
aia', 'container': {'vmid': 102, 'hostname': 'server102.belajar.local', 'ip_address': '1
92.168.0.102/24'}})
skipping: [pve] => (item={'nis': 2503, 'nama_depan': 'Marc', 'nama_belakang': 'Marquez',
  'container': {'vmid': 103, 'hostname': 'server103.belajar.local', 'ip_address': '192.168.0.103/24'}})
skipping: [pve] => (item={'nis': 2504, 'nama_depan': 'Ai', 'nama_belakang': 'Ogura', 'co
ntainer': {'vmid': 104, 'hostname': 'server104.belajar.local', 'ip_address': '192.168.0.
104/24'}})
skipping: [pve]

PLAY RECAP *****
pve                : ok=26   changed=12   unreachable=0   failed=0   skipped=4
   rescued=0   ignored=0
```

MEMVERIFIKASI HASIL PENGHAPUSAN PERMISSIONS

```
ansible servers -m command -a "pvesh get /access/acl"
```

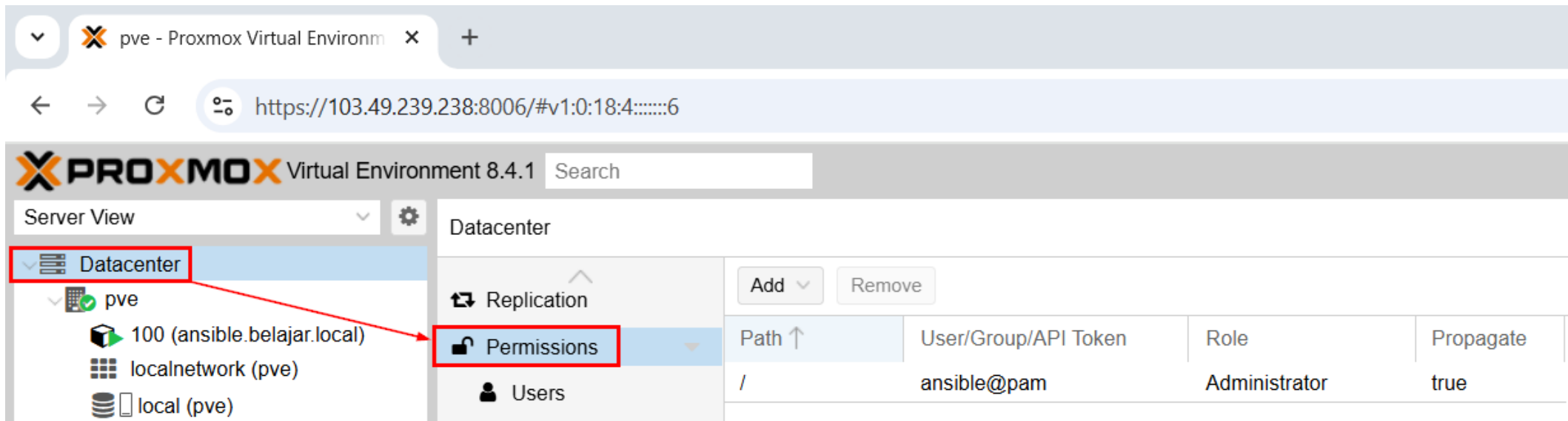
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /access/acl"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at
/usr/bin/python3.11, but future installation of another Python interpreter could change
the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

path	roleid	type	ugid	propagate
/	Administrator	user	ansible@pam	1

Terlihat ijin akses untuk setiap *user* terhadap setiap objek yang telah ditetapkan sebelumnya berhasil dihapus.

MEMVERIFIKASI HASIL PENGHAPUSAN PERMISSIONS

Pada panel sebelah kiri dari **Web GUI PVE**, pilih **Datacenter** kemudian pada panel detail sebelah kanan pilih **Permissions**.



The screenshot shows the Proxmox Web GUI interface. The left sidebar has a 'Datacenter' menu item highlighted with a red box. A red arrow points from this menu item to the 'Permissions' menu item in the 'Datacenter' panel, which is also highlighted with a red box. The 'Permissions' panel displays a table with the following data:

Path ↑	User/Group/API Token	Role	Propagate
/	ansible@pam	Administrator	true

Terlihat ijin akses untuk setiap *user* terhadap setiap objek yang telah ditetapkan sebelumnya berhasil dihapus.

MEMVERIFIKASI HASIL PENGHAPUSAN USERS

```
ansible servers -m command -a "pvesh get /access/users"
```

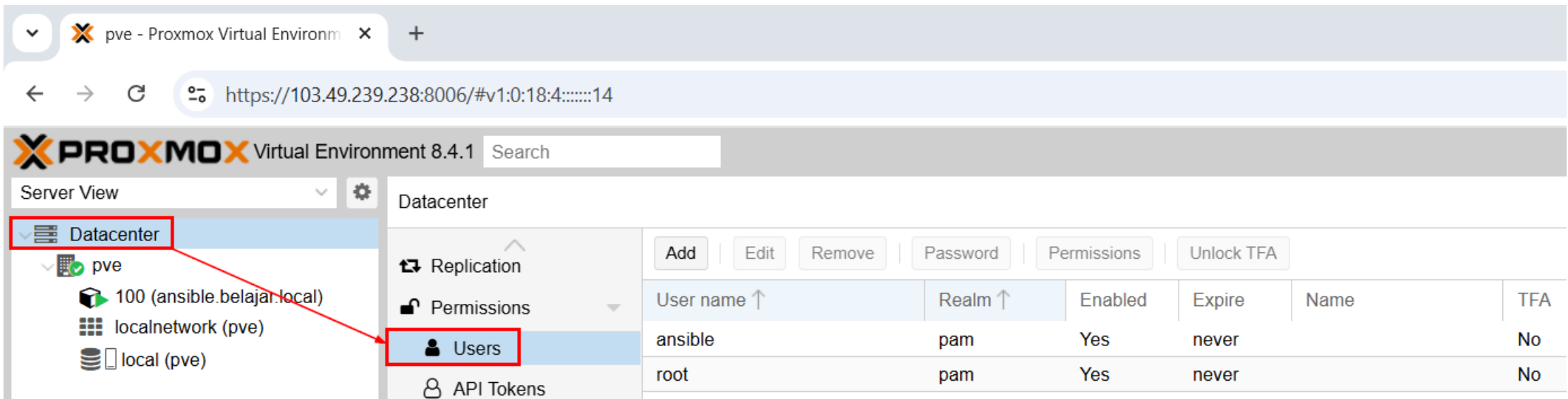
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /access/users"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future
installation of another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

userid	comment	email	enable	expire	firstname	groups	keys	lastname	realm-type	tfa-locked-until	tokens	totp-locked
ansible@pam			1	0					pam			
root@pam			1	0					pam			

Terlihat *user* dengan **user name 2501-2504** telah berhasil dihapus.

MEMVERIFIKASI HASIL PENGHAPUSAN USERS

Pada panel sebelah kiri dari **Web GUI PVE**, pilih **Datacenter** kemudian pada panel detail sebelah kanan pilih **Permissions** → **Users**. Terlihat *user* dengan **user name** **2501-2504** telah berhasil dihapus.



The screenshot shows the Proxmox Web GUI interface. On the left sidebar, the 'Datacenter' menu item is highlighted with a red box. A red arrow points from this box to the 'Users' menu item in the 'Datacenter' panel, which is also highlighted with a red box. The main panel displays the 'Users' list with the following table:

User name ↑	Realm ↑	Enabled	Expire	Name	TFA
ansible	pam	Yes	never		No
root	pam	Yes	never		No

MEMVERIFIKASI HASIL PENGHAPUSAN CONTAINERS

```
ansible servers -m command -a "pvesh get /nodes/pve/lxc"
```

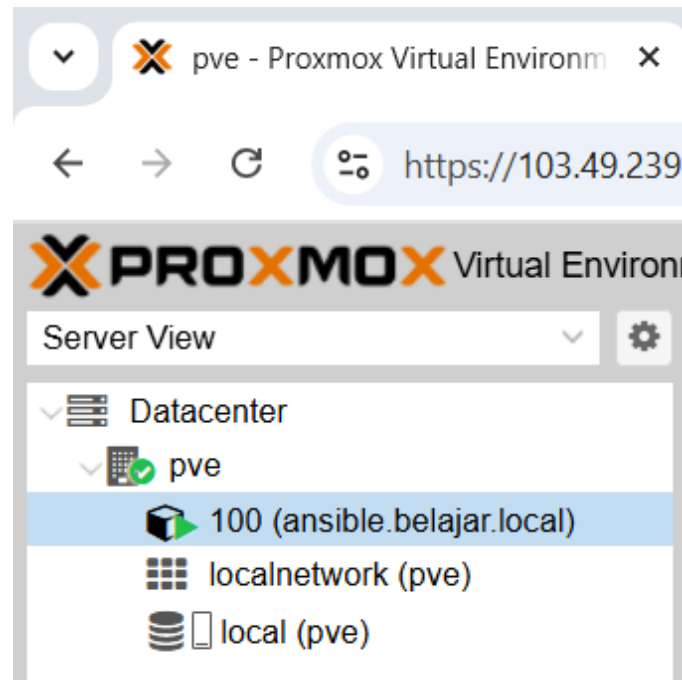
```
root@ansible:~/proxmox-iac# ansible servers -m command -a "pvesh get /nodes/pve/lxc"
[WARNING]: Platform linux on host pve is using the discovered Python interpreter at /usr/bin/python3.11, but future installation of another Python interpreter could
change the meaning of that path. See https://docs.ansible.com/ansible-core/2.18/reference_appendices/interpreter_discovery.html for more information.
pve | CHANGED | rc=0 >>
```

status	vmid	cpu	cpus	disk	diskread	diskwrite	lock	maxdisk	maxmem	maxswap	mem	name	netin	netout	tags	template	uptime
running	100	0	1	1.41 GiB	693.36 MiB	294.40 MiB		3.86 GiB	1.00 GiB	1.00 GiB	131.94 MiB	ansible.belajar.local	5.45 MiB	5.30 MiB			3h 24m 48s

Terlihat **CT ID 101-104** telah berhasil dihapus.

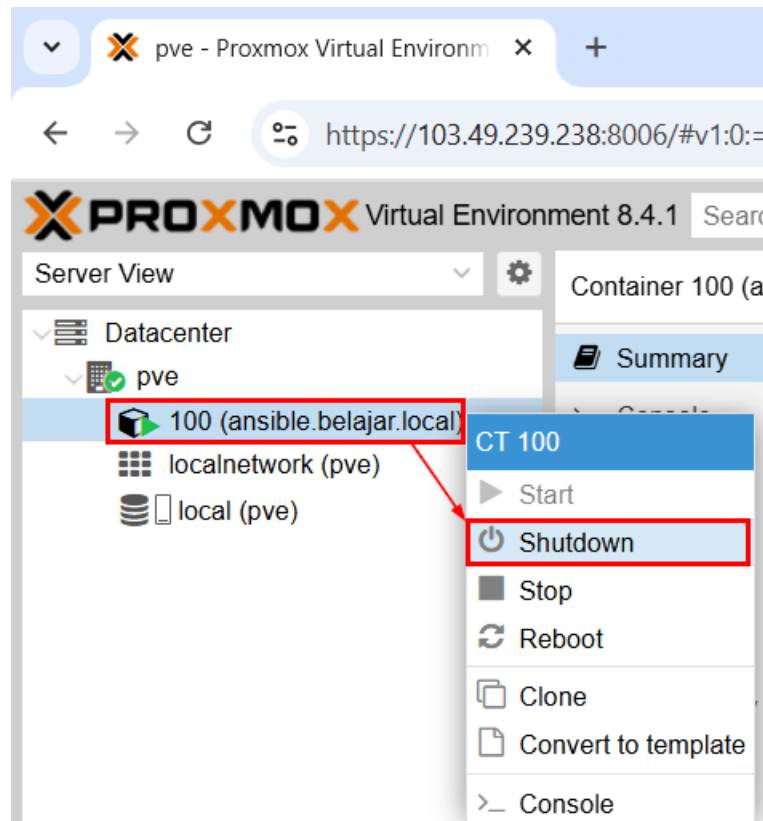
MEMVERIFIKASI HASIL PENGHAPUSAN CONTAINERS

Pada panel sebelah kiri dari **Web GUI PVE** terlihat **CT ID 101-104** telah berhasil dihapus.



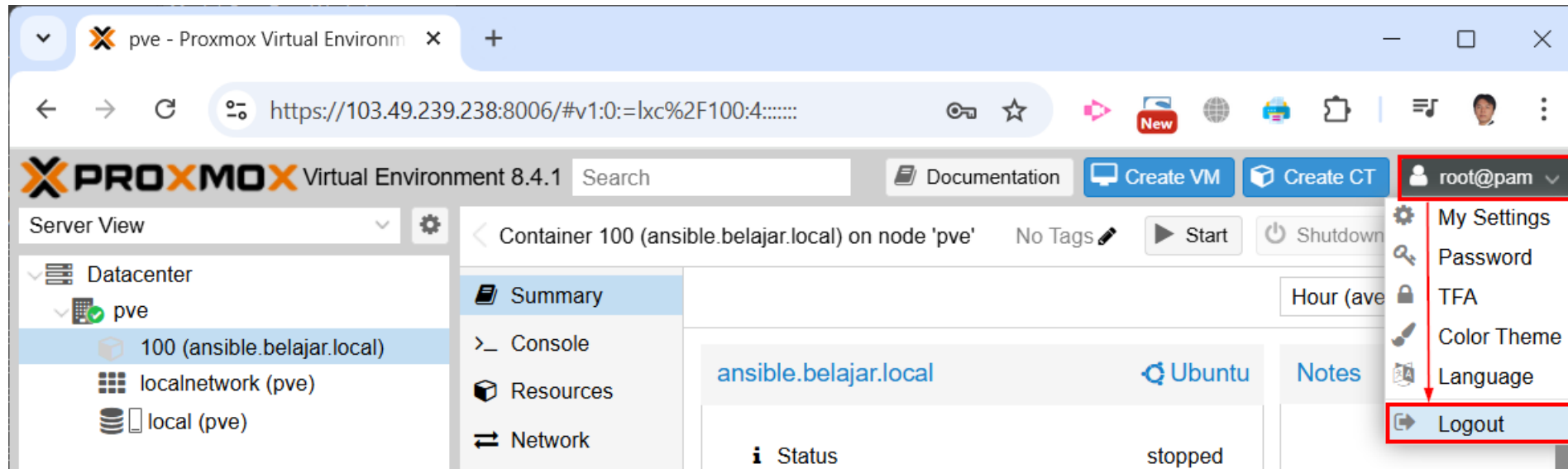
MENGHENTIKAN CONTAINER ANSIBLE.BELAJAR.LOCAL

Pada panel sebelah kiri dari *Web GUI PVE*, pilih **CT “100 (ansible.belajar.local)”** yang terdapat dibawah *node pve* dari menu **Datacenter**. Selanjutnya lakukan klik kanan pada **“100 (ansible.belajar.local)”** dan pilih **Shutdown**.



KELUAR DARI PENGGUNAAN WEB GUI PVE

Untuk keluar dari penggunaan Web GUI PVE dapat dilakukan dengan memilih *dropdown* menu **root@pam** yang terdapat pada pojok kanan atas dan pilih **Logout**.



REFERENSI

- Red Hat, 2025, Understanding Virtualization, <https://www.redhat.com/en/topics/virtualization>
- Red Hat, 2023, What is a Hypervisor, <https://www.redhat.com/en/topics/virtualization/what-is-a-hypervisor>
- Red Hat, 2022, What is Infrastructure as Code (IaC)?, <https://www.redhat.com/en/topics/automation/what-is-infrastructure-as-code-iac>
- Proxmox, 2025, Wiki Proxmox, https://pve.proxmox.com/wiki/Main_Page
- Proxmox, 2025, Proxmox VE Documentation, <https://pve.proxmox.com/pve-docs/>
- Ansible, 2025, Ansible Documentation, <https://docs.ansible.com/>
- Edureka, What is Ansible? – Configuration Management And Automation With Ansible, 2024, <https://www.edureka.co/blog/what-is-ansible/>

ADA PERTANYAAN?

TERIMAKASIH