

Ubuntu Server/Desktop 22.04 / 24.04 / 26.04 Pre- Configuration Guide for High-Load VMS

This guide describes the recommended initial configuration of Ubuntu systems for stable operation of a high-load video surveillance platform based on **Cluebase VMS** running in Docker containers.

The recommendations apply to:

- Ubuntu Server/Desktop 22.04 LTS
- Ubuntu Server/Desktop 24.04 LTS
- Ubuntu Server/Desktop 26.04 LTS

The system is expected to operate under:

- High network throughput
- Intensive disk I/O
- Large numbers of simultaneous video streams
- NVIDIA GPU acceleration
- Continuous 24/7 workloads

1. Fully Update the System

Update package indexes and install all available updates.

```
sudo apt update && sudo apt dist-upgrade -y && sudo apt autoremove -y && sudo apt autoclean &&  
sudo snap refresh
```

Reboot after updates:

```
sudo reboot
```

2. Disable Automatic System Updates

Automatic updates may restart services unexpectedly and affect VMS stability.

Disable unattended upgrades:

```
sudo systemctl stop unattended-upgrades && sudo systemctl disable unattended-upgrades
```

Disable periodic APT updates:

```
sudo nano /etc/apt/apt.conf.d/20auto-upgrades
```

Set:

```
APT::Periodic::Update-Package-Lists "0";  
APT::Periodic::Unattended-Upgrade "0";
```

Save the file.

3. Install Additional System Packages

Install useful administration and monitoring utilities:

```
sudo apt install -y \  
mc \  
htop \  
curl \  

```

```
unzip \  
apt-transport-https \  
ca-certificates \  
gnupg \  
lsb-release \  
gnupg2 \  
iotop \  
net-tools \  
iftop \  
nvme-cli \  
smartmontools \  
jq
```

Package purpose:

Package	Purpose
mc	File manager
htop	Process monitoring
iotop	Disk I/O monitoring
iftop	Network traffic monitoring
smartmontools	Disk health monitoring
nvme-cli	NVMe SSD management

4. Check Secure Boot Status

Secure Boot must be disabled in BIOS/UEFI before installing proprietary NVIDIA drivers.

Check status:

```
mokutil --sb-state
```

Possible output:

```
SecureBoot enabled
```

or

```
SecureBoot disabled
```

If Secure Boot is enabled:

1. Reboot the server
2. Enter BIOS/UEFI
3. Disable:
 - Secure Boot
 - Fast Boot (recommended)
4. Save settings and reboot

5. Install NVIDIA Driver and NVIDIA Container Toolkit

Install NVIDIA drivers only if the server has an NVIDIA GPU.

Recommended driver installation:

```
sudo ubuntu-drivers autoinstall
```

Reboot:

```
sudo reboot
```

Verify driver installation:

```
nvidia-smi
```

Expected result:

- GPU model displayed
 - Driver version visible
 - No errors
-

Install NVIDIA Container Toolkit only after installing Cluebase VMS or Docker

Official documentation:

[NVIDIA Container Toolkit Install Guide](#)

Configure the production repository:

```
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o  
/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg \  
&& curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-  
toolkit.list | \  
    sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-  
keyring.gpg] https://#g' | \  
    sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list
```

Update the packages list from the repository:

```
sudo apt update
```

Install the NVIDIA Container Toolkit packages:

```
export NVIDIA_CONTAINER_TOOLKIT_VERSION=1.19.0-1  
sudo apt-get install -y \  
    nvidia-container-toolkit=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    nvidia-container-toolkit-base=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    libnvidia-container-tools=${NVIDIA_CONTAINER_TOOLKIT_VERSION} \  
    libnvidia-container1=${NVIDIA_CONTAINER_TOOLKIT_VERSION}
```

Configure runtime:

```
sudo nvidia-ctk runtime configure --runtime=docker
```

Restart Docker:

```
sudo systemctl restart docker
```

6. Enable TCP BBR Congestion Control

BBR improves network throughput and reduces latency under heavy traffic.

Open sysctl configuration:

```
sudo nano /etc/sysctl.conf
```

Add:

```
net.core.default_qdisc=fq
net.ipv4.tcp_congestion_control=bbr
```

Apply settings:

```
sudo sysctl -p
```

Verify:

```
sysctl net.ipv4.tcp_congestion_control
```

Expected output:

```
net.ipv4.tcp_congestion_control = bbr
```

7. Configure Static IP Address

Ubuntu 22.04/24.04/24.06 uses Netplan.

Find interface name:

```
ip a
```

Example interface:

```
ens18
```

Edit Netplan configuration:

```
sudo nano /etc/netplan/01-netcfg.yaml
```

Example configuration:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    ens18:
      dhcp4: false
      addresses:
        - 192.168.1.100/24
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses:
          - 1.1.1.1
          - 8.8.8.8
```

Apply configuration:

```
sudo netplan apply
```

Verify:

```
ip a
ip route
```

8. Configure Additional fs.aio-max-nr Parameter

Increase asynchronous I/O limits for high-load applications.

Edit sysctl configuration:

```
sudo nano /etc/sysctl.d/99-aio.conf
```

Add:

```
fs.aio-max-nr = 1048576
```

Apply:

```
sudo sysctl --system
```

Verify:

```
sysctl fs.aio-max-nr
```

Additional Recommended Optimizations for High-Load Cluebase VMS Systems

Increase File Descriptor Limits

Large video systems require high open-file limits.

Edit:

```
sudo nano /etc/security/limits.conf
```

Add:

```
* soft nfile 1048576
* hard nfile 1048576
root soft nfile 1048576
root hard nfile 1048576
```

Also configure systemd:

```
sudo nano /etc/systemd/system.conf
```

Add or modify:

```
DefaultLimitNOFILE=1048576
```

Edit:

```
sudo nano /etc/systemd/user.conf
```

Add:

```
DefaultLimitNOFILE=1048576
```

Reload:

```
sudo systemctl daemon-reexec
```

Increase Network Buffers

Edit:

```
sudo nano /etc/sysctl.conf
```

Add:

```
net.core.rmem_max = 67108864
net.core.wmem_max = 67108864
net.core.rmem_default = 262144
net.core.wmem_default = 262144
net.ipv4.tcp_rmem = 4096 87380 67108864
net.ipv4.tcp_wmem = 4096 65536 67108864
net.core.netdev_max_backlog = 250000
```

Apply:

```
sudo sysctl -p
```

Set Performance CPU Governor

Recommended for dedicated VMS servers.

Install tools:

```
sudo apt install -y linux-tools-common linux-tools-generic
```

Set performance mode:

```
sudo cpupower frequency-set -g performance
```

Verify:

```
cpupower frequency-info
```

Enable TRIM for SSD/NVMe

```
sudo systemctl enable fstrim.timer
sudo systemctl start fstrim.timer
```

Verify:

```
systemctl status fstrim.timer
```

Recommended BIOS Settings

For stable 24/7 operation:

Disable:

- Secure Boot
- Fast Boot
- CPU C-States (optional for latency-sensitive systems)
- ASPM power saving (optional)

Enable:

- Above 4G Decoding (for large GPUs)
 - Resizable BAR (if supported)
 - Performance power profile
-

Monitoring Recommendations

Useful commands:

CPU:

```
htop
```

Disk I/O:

```
iotop
```

GPU:

```
watch -n 1 nvidia-smi
```

Network:

```
iftop
```

Disk health:

```
sudo smartctl -a /dev/sda
```

Final Recommendations

For production Cluebase VMS servers:

- Use Ubuntu Server instead of Desktop whenever possible
- Use enterprise-grade SSD/NVMe drives
- Use RAID with battery-backed cache for archive storage
- Use dedicated NICs for camera traffic
- Prefer 10G networking for large installations
- Keep OS and archive storage separated
- Avoid installing unnecessary GUI applications and services
- Reboot only during maintenance windows
- Use UPS power protection
- Monitor disk temperatures and SMART status regularly

Revision #3

Created 2026-05-21 20:07:36 UTC by Zalim

Updated 2026-05-21 20:27:19 UTC by Zalim