

# Hyperconverged Cloud Architecture with OpenNebula and StorPool

Version 4.0 – February 2026

## Abstract

The Hyperconverged Cloud Architecture with [OpenNebula](#) and [StorPool](#) provides a blueprint to aid IT architects, consultants, administrators, and field practitioners in designing and deploying a hyperconverged cloud infrastructure. In this architecture, nodes combine computing, storage, and virtual networking functions to achieve better efficiency and simplified management. The Hyperconverged Cloud Architecture is a helpful guide for companies seeking to launch fast, easy-to-use, and cost-effective clouds. This document describes the architecture of a complete IT infrastructure, including storage, computing, networking, and cloud orchestration. There are additional components in the open cloud ecosystem that are not part of the reference architecture but are nonetheless essential to consider when designing a cloud, such as configuration management and automation tools for managing cloud infrastructure and large numbers of devices.

## Contents

1. Executive Summary
2. What is OpenNebula?
3. What is StorPool?
4. Benefits of a Hyperconverged Cloud
5. High-Level Reference Architecture
6. Cloud Orchestration
7. Hyperconverged Nodes
8. High Availability
9. Conclusions

## Glossary

ACL	Access Control List
AD	Active Directory
DC	Datacenter
HCI	Hyperconverged Infrastructure
ONE	OpenNebula
QCOW	QEMU Copy on Write format
SAN	Storage Area Network
vCPU	Virtual CPU
VDC	Virtual Datacenters
VM	Virtual Machine

## 1. Executive Summary

This reference architecture provides a proven blueprint for building a scalable and high-performance hyperconverged cloud using OpenNebula and StorPool. It helps enterprises consolidate compute and storage, automate lifecycle management, and reduce costs while preserving flexibility to scale linearly or selectively across hybrid and edge environments.

## 2. What is OpenNebula?

Enterprise cloud computing is the next step in the evolution of data center (DC) virtualization. OpenNebula is a simple, feature-rich, and flexible solution for building and managing enterprise clouds. OpenNebula enables businesses to adopt private, hybrid, and edge cloud computing, meeting the evolving needs of developers for new development tools and frameworks, such as containers or Infrastructure as Code, while ensuring compliance with enterprise requirements for DevOps practices. It combines existing virtualization and containerization technologies with advanced features for multi-tenancy, automatic provisioning, and elasticity.

OpenNebula is a single fully open-source product with a healthy and active community that is commercially supported by OpenNebula Systems. The development of OpenNebula follows a bottom-up approach driven by the real needs of sysadmins, devops, developers, and users. OpenNebula releases are produced regularly and delivered as a single package with a smooth migration path. For more information on the benefits of running an OpenNebula cloud, please visit OpenNebula's discover page.<sup>1</sup>

## 3. What is StorPool Storage?

StorPool Storage is a software-defined storage solution designed for building high-performance public and private clouds.<sup>2</sup> StorPool is block storage software that turns standard servers with direct-attached storage drives into an outstanding storage system, acting as a shared storage pool. Compared to traditional SANs, all-flash arrays, and other SDS products, StorPool is faster, more reliable, and more scalable due to its distributed architecture. StorPool is designed from the ground up to provide cloud builders with the fastest and most resource-efficient storage software in the market. It is tailored for companies building public or private clouds. It provides you with fast and reliable shared storage for your cloud, while significantly reducing the total cost of building and operating a cloud.

## 4. Benefits of a Hyperconverged Cloud

The main goals when deploying a hyperconverged cloud are to gain simplicity, performance, and efficiency without sacrificing features or reliability. The benefits of the proposed hyperconverged cloud architecture are as follows:

- Optimized infrastructure costs: Hyperconverged infrastructure provides a low resource profile without sacrificing performance levels or centralized control. The hyperconverged cloud reduces the need for expensive, specialized staff because it is easy to manage.

---

<sup>1</sup> <https://opennebula.io/discover/>

<sup>2</sup> <https://storpool.com/get-started>

- Better performance and reliability: A well-designed hyperconverged infrastructure has a lot of performance available to be shared by all applications. In the hyperconverged concept, there is usually one fast tier of both storage and computing, so all applications benefit.
- Simplified management: The fewer systems you have to manage, the easier it will be for your company. Especially if you do not have a huge team of dedicated system administrators or infrastructure engineers.
- Simplified hardware inventory: Repeated identical building blocks.
- Scalability: Scale compute and storage simultaneously. Capacity and performance can be scaled together by adding a new server to the cluster. Each VM can be scaled up to very large CPU, RAM, storage capacity, and performance.

## 5. High-Level Reference Architecture

A cloud architecture is defined by components in four planes: cloud orchestration, computing, storage, and networking. Figure 1 shows an overview of a hyperconverged cloud using OpenNebula and StorPool Storage. OpenNebula services run in the cloud orchestration plane and communicate with other components over the Service Network. OpenNebula services monitor the status of the hypervisors and Virtual Machines (VMs), and initiate VM- or storage-related operations.

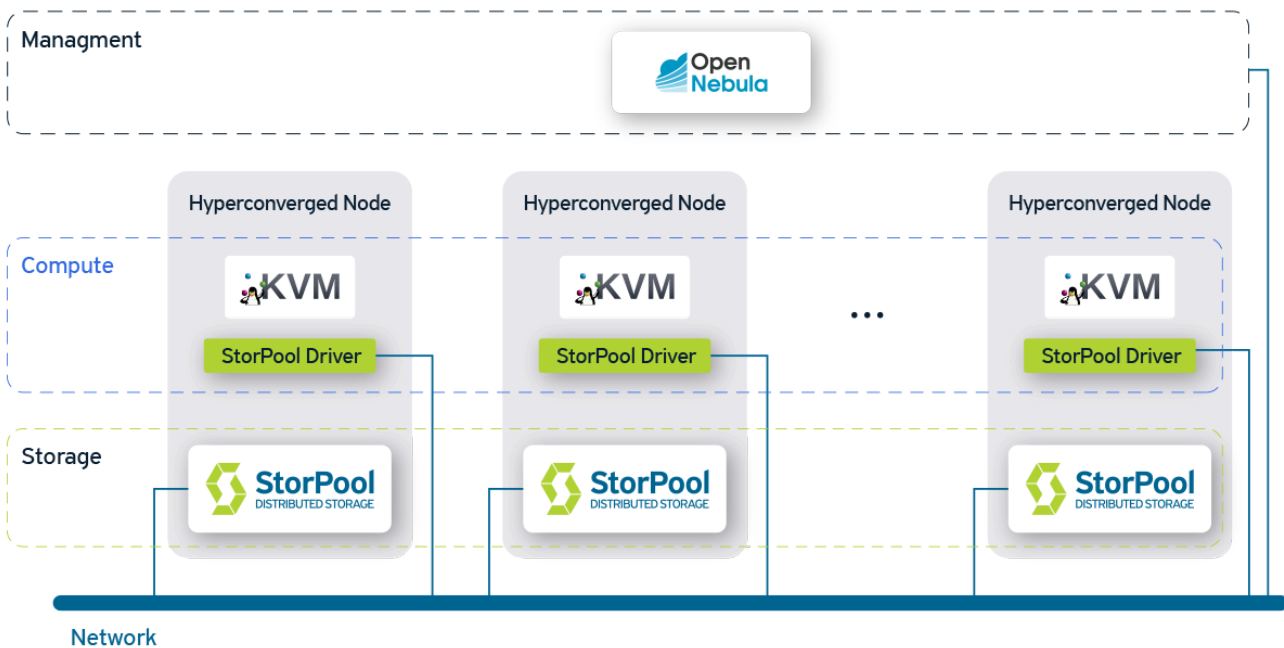


Figure 1. Hyperconverged Architecture

The hypervisors have identical configurations running the same set of components in all planes. This architecture provides efficient resource utilization, a high level of redundancy, and simplified management. Each node contains CPU, memory, storage, and networking resources, allowing them to act as storage and compute nodes simultaneously. Each node runs the KVM hypervisor, StorPool Storage software, and the OpenNebula drivers to manage local resources. Each hypervisor is connected to all three networks: public and private networks for user connections, service, and storage networks.

The cloud grants virtual machines access to dedicated virtual private networks (VPNs), allowing for closed, isolated communication between only a specified set of virtual machines, or to a shared public virtual

network. Access to each virtual network can be restricted to different users or groups, and/or have different QoS settings and rate limits applied to them. The Public Network is used to connect VMs to the Internet.

Table 1 provides a simple summary of the various components used to create this architecture.

Authentication	Native authentication or Active Directory
Management	OpenNebula
Hypervisor	Linux KVM
Networking	Redundant 10+Gbps storage network, 1+Gbps network shared by service network, public and private virtual networks
Storage	StorPool Storage
Operating System	Supported Linux distribution (Ubuntu, RHEL/RHEL Clones like AlmaLinux) on all hosts

**Table 1.** Summary of the implementation

## 6. Cloud Orchestration

All resources of the hyperconverged cloud are orchestrated by OpenNebula front-ends. In the Hyperconverged Architecture, all OpenNebula front-ends are running in virtual machines. There is an option for one of the OpenNebula front-ends to be on a physical server instead. The recommended operating systems for the front-end are RHEL, AlmaLinux, and Ubuntu. The VM sizing/hardware recommendations are listed in Table 2. These recommendations are intended solely as guidance.

Memory	8 GB
CPU	2 CPU (4 cores)
Disk size	200 GB
Network	2 NICs

**Table 2.** Front-end hardware recommendations

The front-end provides the following services:

- OpenNebula management daemon
- Scheduler
- MySQL DB
- Administration and User GUI and APIs
- Optional OpenNebula services like OneFlow or OneGate

OpenNebula front-ends rely on a distributed consensus protocol to provide fault-tolerance and state consistency across OpenNebula services

## 7. Hyperconverged Nodes

In a hyperconverged cloud architecture, each node implements all cloud functions: Virtualization, Storage, and Networking. This provides a high level of availability with optimum resource utilization. An example of hyperconverged node architecture is shown above in Section 5 in Figure 1. The various elements of this architecture are described in more detail below.

### 7.1. Compute

Compute nodes are responsible for providing VMs with execution resources such as CPU, memory, and network access. The recommended operating systems for the virtualization nodes are RHEL, AlmaLinux, and Ubuntu. The recommended hypervisor in the architecture is the KVM delivered by the platform OS. The configuration of the nodes is homogeneous in terms of the software components installed, the oadmin administration user and accessible storage. To keep the cloud simple and efficient, configure a smaller number of nodes with a high number of CPU cores.

One of the most important tasks defining a cloud infrastructure is to dimension the virtualization nodes according to the expected workload. A typical virtualization node comes with approximately 384 GB RAM and 40 CPU cores (80 logical CPUs). This equates to 4.8 GiB RAM per logical CPU. Depending on the expected workload and its requirements in terms of memory vs CPU, as well as resource oversubscription, the optimal amount of RAM per node might be smaller or larger. For example, if the requirement is for 2 GB RAM per vCPU and there will be no vCPU oversubscription, a hypervisor with 80 logical CPUs would need only 160-192 GB RAM. In any case using larger, denser servers like the ones described here is usually much more efficient than using servers with a much smaller number of CPU cores.

### 7.2. Storage

Storage is one of the most critical aspects of a cloud infrastructure, and needs to be carefully planned to avoid bottlenecks.

StorPool distributed storage presents unique characteristics, delivering consistently high performance, low latency, and availability using a small and fixed amount of resources. StorPool is built using a distributed, shared-nothing architecture. All functions are performed by all nodes on an equal peer basis. StorPool scales linearly in capacity and performance. The bigger the cluster grows, the higher the performance and throughput it can provide.

The StorPool software that is installed on each node (host) consists of two parts: a storage server service and a storage client/driver service. All storage servers and clients are connected over a dedicated redundant 10G-100G storage network for minimal latency and maximum throughput.

StorPool servers on all nodes collaborate to create a distributed service, where each StorPool server stores part of the user data. Redundancy is guaranteed through a synchronous replication algorithm. This can be thought of as a very advanced software RAID between servers. By combining all available drives of all servers in a single storage pool, the user is not limited by the capacity or performance of a single disk. Every volume, virtual disk or OpenNebula image, is distributed on multiple storage nodes and multiple disks, ensuring resilience against server, disk, or network failures.

The StorPool client communicates in parallel with all StorPool servers to achieve the aggregate throughput and IOPS of all installed disks in the cluster. Volumes defined in the storage system are presented by the StorPool client as standard block devices to the operating system.

With the OpenNebula integration, every virtual disk or image corresponds to a volume in StorPool that is presented as a block device in the host where it is used. Hence, virtual machine images are stored directly in the StorPool volumes using the "raw" format, without a filesystem or a VM image format, such as QCOW2,

as an extra layer, reducing the overhead and maximizing the performance. All operations on the VM images, such as creating snapshots, cloning images, and thin provisioning, are implemented natively by StorPool.

StorPool volume management is integrated with OpenNebula to allow seamless use of the capabilities of the storage system through OpenNebula GUI, CLI, and API interfaces. The result of the deep integration between the two systems is that it allows for automation of all tasks in the storage system.

The deep integration between OpenNebula and StorPool allows efficient use of the features of the storage system. For example, when a new virtual machine is instantiated from a template, the virtual disk is created as a clone of the master image. Being a copy-on-write system, StorPool creates clones without consuming additional redundant space. This can result in significant savings in large clusters with many identical virtual machines.

### 7.3. Networking

In terms of networking, configure four network interfaces per node. Two interfaces used for the storage functions and two additional interfaces used for the public, private and service networks.

### 7.4. Scalability and Growth Models

With hyperconverged infrastructure all resources scale together linearly. When a new host is added to the cluster, compute, network, and storage resources scale simultaneously.

However, not all workloads are created equal. It is possible that linear scaling might not be cost-effective for specific workloads. For example, some workloads might need to scale up compute and network resources, but have no need for additional storage resources. Likewise, some workloads might need to scale up storage but have no need for additional compute and network resources.

With standard hyperconverged implementations, scaling all resources linearly and simultaneously is the only option available.

StorPool’s distributed storage offers more flexibility for future scaling. With StorPool, it is simple to add either compute/network only nodes – or storage-only nodes – and integrate them into the cluster with the existing hyperconverged nodes. An example of this mixed infrastructure architecture is shown in Figure 2 below.

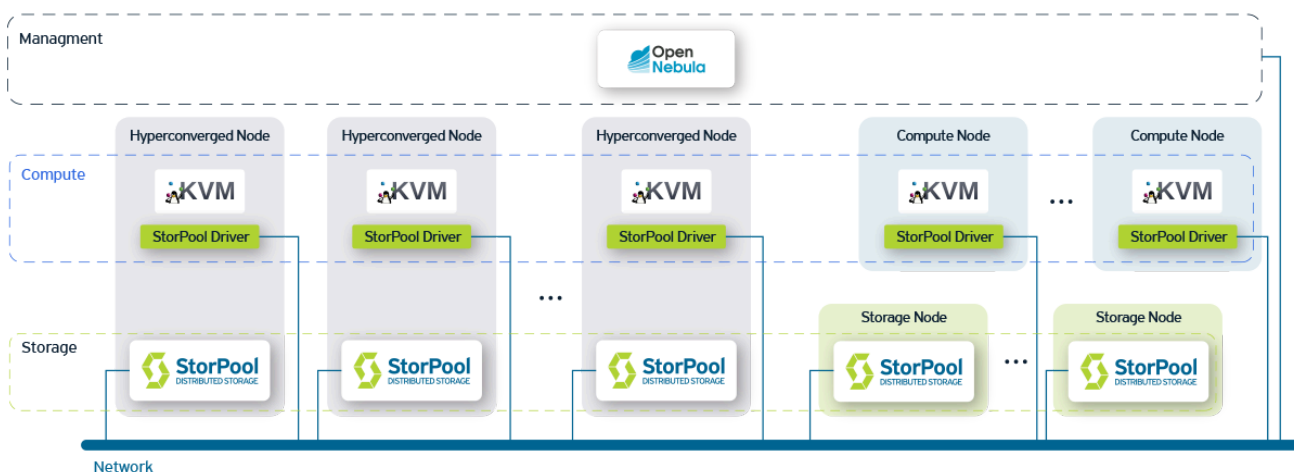


Figure 2. Mixed Infrastructure Architecture

This flexibility allows for future-proofing the infrastructure, as one is able to scale the cluster linearly and/or non-linearly to meet changing future growth needs.

## 8. High Availability

High-availability design is implemented in each plane of the Hyperconverged Cloud Architecture.

### 8.1. Cloud Orchestration

OpenNebula uses a distributed consensus protocol based on Raft to provide fault-tolerance and state consistency across OpenNebula services.

To preserve a consistent view of the system across servers, modifications to the system state are performed through a special node—the leader. The servers in the OpenNebula cluster elect a single node to be the leader. The leader periodically sends heartbeats to the other servers—the followers—to keep its leadership. If a leader fails to send the heartbeat, followers get promoted to candidates for the leader position, and start a new election. Read-only operations can be performed through any OpenNebula server in the cluster. This means that reads can be arbitrarily stale but generally within the round-trip time of the network. A minimum of three front-end needs to be deployed in order to support one node failure. HA can also be configured for VMs, for instance to be re-launched if they enter a fail state, or it can be configured for virtualization nodes, so all the VMs running in a crashed node get moved automatically to another node.

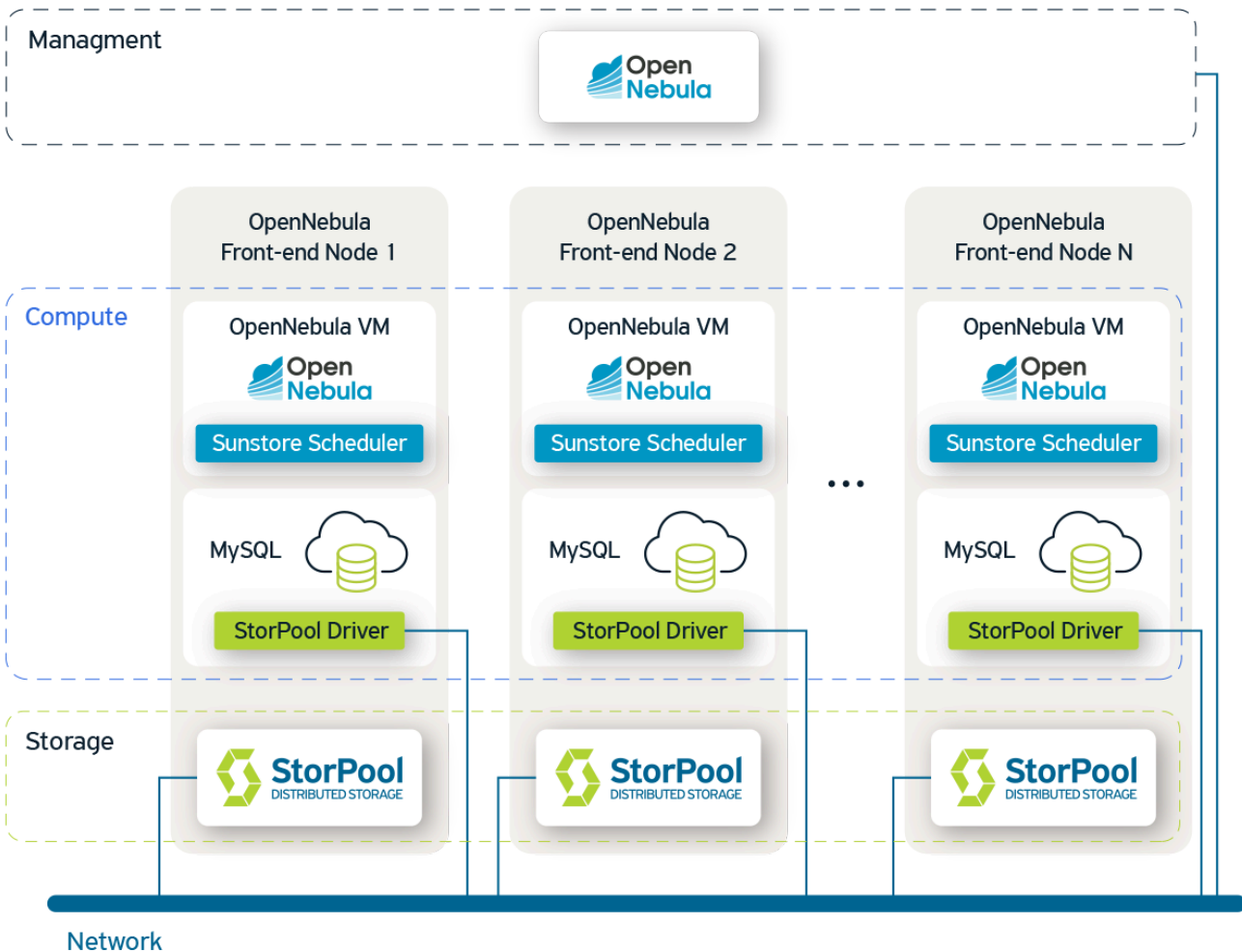


Figure 3. Overview of the OpenNebula HA architecture and main components

## 8.2. Compute plane

In case of a failure of a hypervisor on a host, all VMs on the host are automatically relaunched on remaining operational hypervisors on other hosts, resulting only a short, predictable downtime for the virtual machines. Planned maintenance operations can be performed without service downtime, by using live migration of the virtual machines off the hypervisor to another host, before it is shut down or restarted. This allows hardware or software upgrades of individual nodes or an entire cluster to be performed without any effect on services.

The shared storage provided by StorPool to all of the hosts makes live migrations of VMs simple and fast. This shared storage also allows for handling individual virtual machine failures, automatically relaunching VMs on the same host or a different host as needed.

## 8.3. Storage plane

Storage is the most critical component of the cloud in terms of availability. In the storage plane, high availability is built in several layers.

Disk failures are handled by multiple redundant data copies (usually three), always on different disks and different nodes. In case of a disk failure, StorPool storage will continue normal operation using the remaining active copies of the data.

With end-to-end data integrity and self-healing, StorPool automatically recovers from data loss caused by disk or node failures, bit rot, undetected disk, memory, or network errors, by restoring missing or corrupted copies to preserve the configured level of data redundancy.

Host failures are mitigated by implementing shared-nothing distributed architecture. All the components inside a hypervisor, such as client, server, and others, are replicated across all the available hypervisors to ensure data redundancy and high availability of the system. In case of an issue with a server, the system continues to work uninterrupted, without downtime or loss of data.

Unlike the StorPool server component which uses an active-active load-balancing redundancy scheme, the StorPool API (used for provisioning and monitoring functions) relies on an active-passive redundancy scheme with a floating IP address.

## 8.4. Networking

Hardware nodes operate with redundant physical connections to two separate network switches with automatic fallback mechanisms for host, network card, interface, cable, or switch failure. On the public, private, and service networks, the system uses Linux Bonding to provide a highly-available network connection. Depending on capabilities of the switches, the Bond interface will be configured for LACP (active-active load-balancing) or active-backup mode. On the storage network side, the network architecture uses StorPool's integrated multi-path functionality to provide HA and use both of the interfaces for increased throughput when they are available.

## 9. Conclusions

The Hyperconverged Architecture described in this document has been created from the collective information and experiences from hundreds of users and cloud client engagements to help in the design and deployment of hyperconverged infrastructures. The document recommends software products and configurations for a smooth OpenNebula with StorPool cloud installation. However, in many cases, there are other aspects to be taken into account, like infrastructure platforms and services pre-existing in the datacenter as well as the provisioning processes of the corporation.

## About OpenNebula

Enterprise cloud computing is the next step in the evolution of data center (DC) virtualization. OpenNebula is a simple, feature-rich and flexible solution to build and manage enterprise clouds. OpenNebula enables businesses to embrace private, hybrid and edge cloud computing and meet the constant needs of developers for new development tools and frameworks, like containers or Infrastructure as a Code, while ensuring enterprise requirements for DevOps practices. It combines existing virtualization and containerization technologies with advanced features for multi-tenancy, automatic provisioning, and elasticity.

## About StorPool

StorPool is a software-defined storage solution for building high-performance public and private clouds. It runs on standard servers, drives, and networks and turns them into an outstanding storage system. StorPool is block-storage software installed on the servers, which creates a shared storage pool from the local drives in these servers. Compared to traditional SANs, all-flash arrays, and other SDS products, StorPool is faster, more reliable and more scalable due to its distributed architecture

## LET US HELP YOU DESIGN, BUILD, AND OPERATE YOUR CLOUD



### CONSULTING & ENGINEERING

Our experts will help you design, integrate, build, and operate an OpenNebula cloud infrastructure



### OPENNEBULA SUBSCRIPTION

Get access to our Enterprise Edition, support and exclusive services for Corporate Users



### CLOUD DEPLOYMENT

Focus on your business and let us take care of setting up your OpenNebula cloud infrastructure

Sign up for updates at [OpenNebula.io/getupdated](https://OpenNebula.io/getupdated)

© OpenNebula Systems 2026. This document is not a contractual agreement between any person, company, vendor, or interested party, and OpenNebula Systems. This document is provided for informational purposes only and the information contained herein is subject to change without notice. OpenNebula is a trademark in the European Union and in the United States. All other trademarks are property of their respective owners. All other company and product names and logos may be the subject of intellectual property rights reserved by third parties.

Rev4.0\_20260205