apachecloudstack™
open source cloud computing

Apache CloudStack and StorPool Storage

Solution Brief

1 Executive Summary

Cloud builders around the world - Managed Services Providers, Hosting Services Providers, Cloud Services Providers, Enterprises, and SaaS Vendors - are always seeking ways to optimize the architecture and cost of their public and private cloud deployments. With Apache CloudStack and StorPool Storage, they can deploy a highly automated cloud that delivers the application performance and reliability their users need while easily addressing changes in user requirements over the long term.

This document aims to be a blueprint that aids cloud and storage architects, consultants, administrators, and field practitioners in the design and deployment of reliable, agile, fast, easy to manage, and cost-efficient cloud infrastructure based on Apache CloudStack working with KVM hosts and StorPool Storage. If you need more information and would like to get started with your cloud deployment, get in touch with StorPool Storage.

apachecloudstack
open source cloud computing

2 About Apache CloudStack

Apache CloudStack is the leading open-source cloud orchestration platform used by many of the world's largest public and private clouds. It is a multi-hypervisor, multi-tenant, high-availability Infrastructure as a Service cloud management platform.

Apache CloudStack is a software that provides a cloud orchestration layer, giving automation of the creation, provisioning, and configuration of IaaS components (e.g., virtual machines used for virtual servers, databases, load balancers, virtual desktops, and more). It turns existing virtual infrastructure into a cloud-based Infrastructure as a Service (IaaS) platform. Because CloudStack leverages existing infrastructure, the cost and time for the organization to build a multi-tenant IaaS platform are greatly reduced.

Among the most significant advantages of the virtualization management platform is the simplicity and ease of use it brings, even for large-scale environments. With CloudStack, you can orchestrate hosted public and private clouds, on-premise private clouds, and hybrid environments without the need of engaging a huge operations team to support them in the long term.

About StorPool Storage

StorPool Storage is a primary storage platform designed to be the ideal foundation for high-performance public and private clouds. It is the easiest way to convert sets of commercial off-the-shelf servers with standard storage drives and network devices into high-performance, linearly scalable primary storage systems.

StorPool Storage systems are ideal for storing and managing the data of workloads that demand extreme reliability and low latency - databases, web servers, virtual desktops, real-time analytics, and other mission-critical software. Compared to traditional SANs, all-flash arrays, or other storage software, StorPool Storage is faster, more reliable, and 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 on the market.

Under the hood, StorPool is a block storage software that runs on standard off-the-shelf servers with datacenter-grade storage drives to build out shared-storage pools. StorPool's software comes with a full set of managed services to ensure cloud builders get fully functional fast and reliable storage systems.

StorPool's Integration with Apache CloudStack

StorPool Storage volume management is integrated with Apache CloudStack to allow seamless use of the capabilities of the StorPool Storage system through the CloudStack GUI, CLI, and API interfaces. With the integration, the features available in StorPool get inherited by each cloud deployed with StorPool Storage - enabling cloned provisioning, instant snapshots, thin provisioning, and backup, disaster recovery, and Quality of Service (QoS) policies per virtual disk or virtual machine (VM). Thanks to the way StorPool works, VM provisioning is nearly instantaneous and data placement policies and other settings can be changed in-flight to address changes in user requirements.

The end result of the deep integration between the two systems is that there is no need to manage the storage system directly because all the interactions between CloudStack and StorPool are automated. For example, creating a virtual disk in CloudStack directly triggers the creation of a volume in StorPool. Another example is cloned provisioning - when provisioning a new VM based on a CloudStack template, StorPool clones the template to create a new volume and avoids unnecessary data duplication. At scale, this feature has a significant impact on the cloud's efficiency and significantly optimizes the cost of the storage system.

apachecloudstack open source cloud computing
StorPool DISTRIBUTED STORAGE

Benefits of a Joint Deployment with CloudStack and StorPool Storage

The integration between Apache CloudStack and StorPool Storage results in several business and technical benefits.

Business Benefits:

• **Always online** - The environment does everything online - updates, upgrades, scaling with additional storage or compute servers, hardware replacements, and more. As a result, you can forget about the painful downtime scheduling with customers and the cash drains caused by Service Level Agreement breaches.

• **End-to-end automation** - Having a native integration that enables seamless management of your cloud infrastructure from the CloudStack user interfaces means that your tech teams finally have time for the critical projects that aim to grow your business instead of managing the individual building blocks of your cloud.

• **Operational flexibility** - You can build multiple service offerings with fast, medium, or slow storage tiers tuned for your customers' use cases so that you can deliver the solutions that meet the needs and budgets of various customers. You can also address evolving user requirements by simply changing their workload settings live.

• **Component interoperability** - It is possible to deploy, upgrade, and grow your environment using a mix of cost-efficient standard components suitable for each individual use case.

• **Effective hardware lifecycle management** - You can change between generations of hardware without interrupting your services. Your team will no longer have to forklift amortized equipment and perform slow and complex migration projects every few years.

Technical Benefits:

• **Simplified operations** – reduced time required for operations tasks, faster introduction of technologies, easier employee onboarding and training.

• **Faster deployment** with reduced technological risk and shorter time to market

• **Reduced outages** caused by human errors.

• **Shared continuous storage space** for all hosts in a zone.

• **Independent scaling** of the compute and storage resources.

• **Fast live migration of VMs** from any to any host without data movement.

• **Resilience to hardware failures** as data integrity is maintained upon failure of a drive, storage node, host, or even an entire pod.

• **All storage operations are automated**, eliminating storage administration tasks like volume management, capacity management, etc.

• **Increased service availability** – In case of a host failure, only VMs running on the failed host will be affected. The affected VMs can be recovered in seconds, by automatically restarting them on an available host.

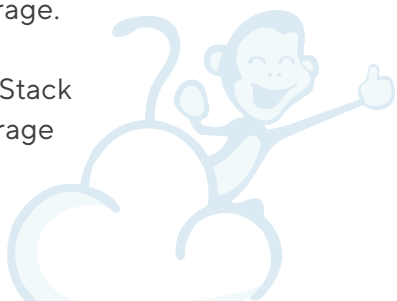
• **Multiple storage tiers** served from the same primary storage can be made available to all pods and hosts.

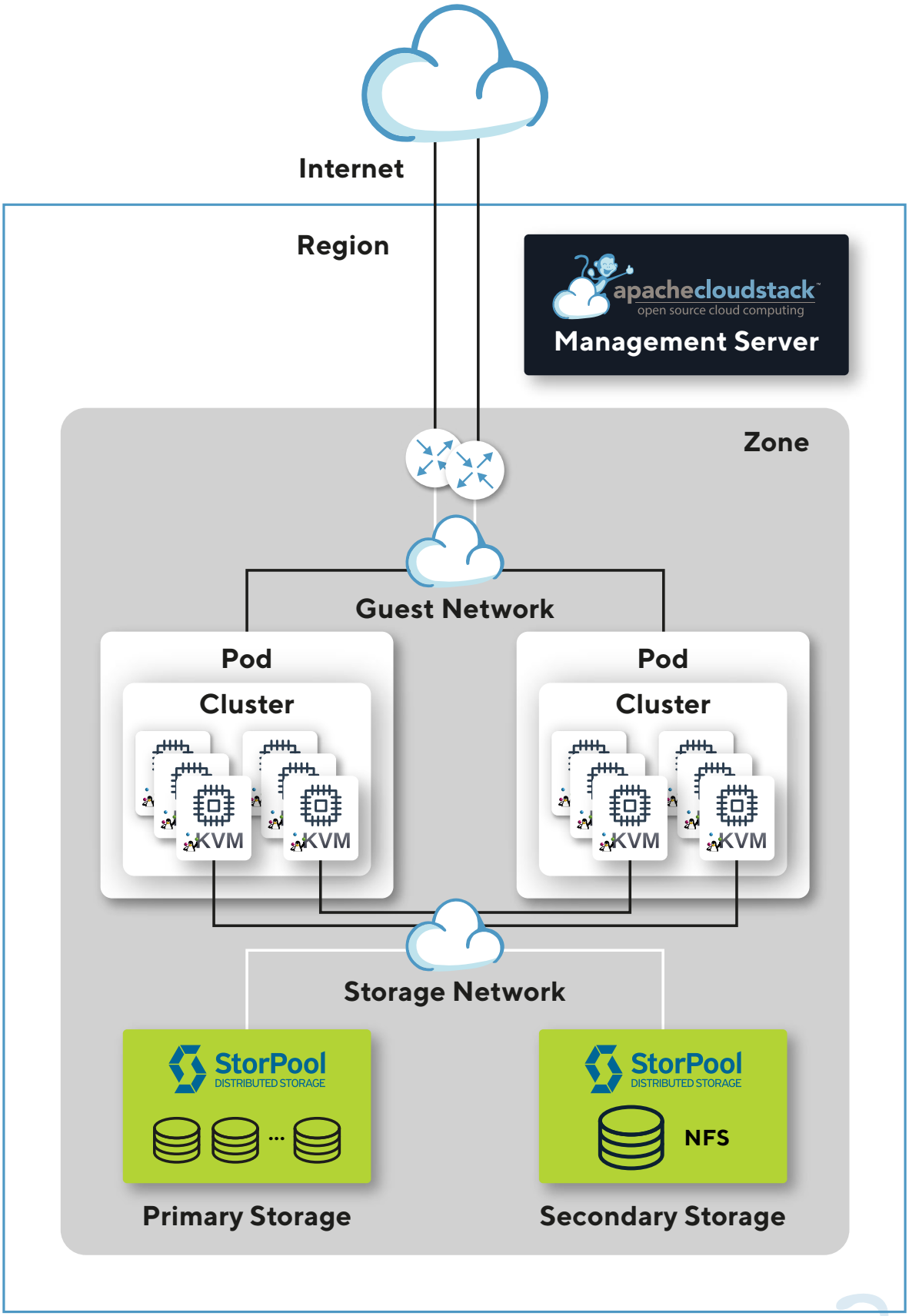
• **Simplified cloud architecture** since both primary and secondary storage can be provided with a single storage system.

High-level Reference Architecture

While every cloud is unique, this document presents some common design principles that can help readers plan how to build their cloud with Apache CloudStack and StorPool Storage.

The main components included in the described solution are the Apache CloudStack cloud management platform, KVM hosts, primary block storage by StorPool Storage and (optionally) secondary NFS storage by StorPool Storage.




Internet
Region
apachecloudstack
open source cloud computing
Management Server
Zone
Guest Network
Pod
Cluster
KVM
KVM
Pod
Cluster
KVM
KVM
Storage Network
StorPool
DISTRIBUTED STORAGE
Primary Storage
StorPool
DISTRIBUTED STORAGE
NFS
Secondary Storage

Cloud Management

Apache CloudStack relies on several organizational units to group and manage resources - regions, zones, pods, and clusters. This gives great flexibility in the way the cloud is built to cover diverse use cases - from small private clouds to multi-regional public clouds.

The presented solution is built with a single zone and one or more pods and clusters of KVM hosts. It can scale to multiple zones managed by the same CloudStack Management Server by multiplicating the zone configuration.

The CloudStack Management Server controls the KVM hosts, the network, and the StorPool primary storage.

Hosts

Hosts are built with Linux KVM hypervisors. While CloudStack supports many different hypervisor technologies in the same cloud, this solution brief is based on Linux KVM hypervisors.

Hosts are configured **without local storage**. In each server, only a boot disk is required for the operating system and various software components. For the virtual machines, hosts use the **shared block storage** provided by the underlying StorPool Storage system.

Hypervisors are connected to three networks - the management network, the guest network, and the storage network. More details are given in the Network section.

Depending on the scale of the cloud and the number of hosts, they can be grouped in one or more clusters and pods. Usually, one cluster is equivalent to one rack in the data center, but this may vary.

Storage

The Apache CloudStack architecture defines two types of storage - primary and secondary.

Primary Storage

The primary storage stores the virtual disks for the VMs in the zone. In this solution, the primary storage is provided by the StorPool block storage software. The StorPool storage cluster is built out of three or more standard Linux servers with local drives. The StorPool software pools the capacity and the performance of all drives on the storage nodes to provide a continuous storage space for the hosts in the zone. The StorPool storage system can scale up - by adding more drives to the existing storage nodes - or scale out - by adding more storage nodes - without service interruptions. This allows starting with a small initial deployment and growing the capacity and performance of the primary storage when the needs increase.

The StorPool block protocol is used for communication between the hosts and the storage system. It delivers great performance and end-to-end data integrity to ensure that the data you put into the storage system is the data you get back, irrespective of hardware or software mishaps that could occur during writing or over time.

The primary storage is shared by all hosts - any virtual disk can be accessed by any host without copying the data. The zone-wide shared storage enables flexible resource allocation and fast and reliable live migration of VMs between any two hosts. A VM can be started on any host with available CPU and memory resources, unconstrained by the required storage tier or capacity.

With the deep integration between Apache CloudStack and StorPool Storage, every virtual disk is stored as a separate volume in the StorPool storage system. This approach allows any storage-related operation like virtual disk provisioning, snapshots, or cloning to be offloaded and executed efficiently by the primary storage system.

Another advantage of this approach is simplified operation and improved performance by eliminating multiple layers of overhead. Because all the required features are implemented in the storage system, layers like a file system, LVM, or QCOW images are not needed anymore. Each virtual disk is directly mapped to a volume in StorPool that is presented as a raw block device on the hypervisor.

The typical primary storage system is built with SSDs (SATA, SAS, or NVMe) for low latency and maximum performance. StorPool Storage allows combining multiple storage tiers for different types of workloads in the same storage nodes. For example a pool of NVMe SSD devices for tier-one virtual disks and HDDs for volumes storing colder data or backup tiers.

Support for StorPool primary storage is added in Apache CloudStack version 4.17. Earlier versions require installing a storage plugin to integrate with StorPool.

Secondary Storage

The Apache CloudStack architecture requires an NFS secondary storage in each zone. Traditionally, the secondary storage is used for several purposes:

- Storing and providing templates or ISO images when a new VM is created
- Storing snapshots
- Transferring volumes between CloudStack clusters

To deliver these services, the secondary storage requires sufficient capacity and bandwidth.

When a zone-wide StorPool primary storage is deployed, the secondary storage can be configured to only store VM templates. In this case:

- **There is no need to download VM templates for VM deployments** - VM templates are downloaded from the secondary storage on the first use only and cached on the primary storage.

- **No snapshot transfers** – Snapshots are implemented in StorPool Storage and do not need to be stored on the secondary storage system.
- **No volume migration** – The primary storage is shared between all hosts in the zone, and any volume is available to any node. This eliminates the need for transferring large amounts of data.

These reduced requirements for the secondary storage give cloud builders a wider choice. They could either deploy a small NFS server on the CloudStack Management Server, use a third-party NFS storage system, or hand over responsibility for this cloud component to StorPool Storage.

To simplify the solution brief, the recommended cloud design uses the StorPool NFS service, which is also the best choice for cloud builders looking to simplify their cloud architecture. In this deployment option, StorPool Storage provisions a highly-available NFS secondary storage with the StorPool storage system used for primary storage. The NFS storage is backed by the same distributed block storage platform used for primary storage.

Network

The presented solution contains three networks – management network, guest network, and storage network.

Management Network

The management network provides connectivity between the hosts and between the Apache CloudStack Management Server and hosts. In most deployments, the management network uses one or two dedicated 1Gbit/s Ethernet interfaces on the hosts.

Guest Network

The guest network carries the user traffic between virtual machines and inbound/outbound traffic to and from the public Internet. The guest network can be implemented either with layer 2 or layer 3 network and VXLAN. In most cases, the guest network uses two dedicated interfaces per host running at 10Gbit/s or faster.

Primary Storage Network

The primary storage network provides the connectivity between the hosts and the primary storage and between the storage nodes. It is critical for the cloud operations, as all virtual machine data is stored on the shared primary storage. The parameters of the primary storage network like latency and bandwidth directly impact the service quality. Special attention needs to be paid to build a highly available storage network without bottlenecks.

Modern storage networks usually use two dedicated 10Gbit/s or faster Ethernet

interfaces on each host and storage node. StorPool Storage uses multipathing for load-balancing and failover in case of network failures. For maximum availability, the primary storage network is implemented with a redundant set of switches. The network topology depends on the scale of the cloud and the physical topology, but simple configurations are preferred to reduce the complexity and the risk of human errors. In many cases, a simple layer 2 network with a pair of top-of-rack switches is the best option.

In many cases, the guest and the primary storage networks can share the same pair of interfaces on hosts with 25 Gbit/s or faster networks, reducing the number of required high-speed interfaces to two.

Overview of Deployment Modes

The presented solution can be built as Disaggregated Infrastructure - using dedicated compute and storage nodes - or as Hyperconverged Infrastructure (HCI), where every node combines compute and storage functions.

Hyperconverged Infrastructure

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

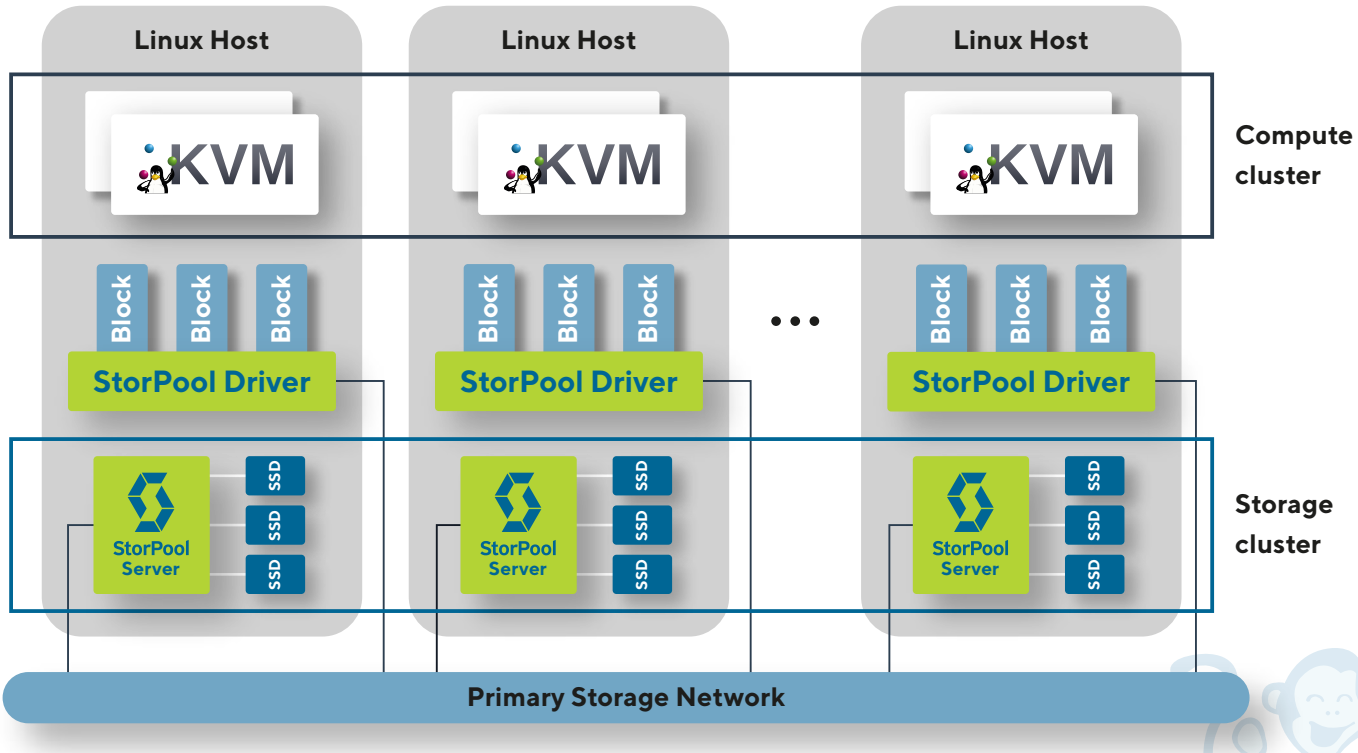
- **Lowest upfront cost** - Since you do not need to buy dedicated storage servers initially and plan for their networking, you can save up to 30% of the start-up cost for hardware.
- **Optimized infrastructure costs** - Hyperconverged infrastructure provides a low-resource profile without sacrificing performance levels or centralized control. The hyperconverged cloud reduces the need for expensive, specialized staff because it is easier to manage.
- **Simplified management** - Fewer systems to manage, fewer network elements, simpler logical, physical, and network topology.
- **Simplified hardware inventory** - Repeated identical building blocks.
- **More straightforward scaling** - It is possible to scale the capacity and performance of the compute and storage together by adding a new server to the cluster. Each VM can be scaled up to a substantial vCPU, RAM, storage size, and performance.
- **Ability to scale up** - With the right hardware platform, cloud builders can scale up the storage and memory capacity when the need arises by adding storage drives or RAM to their existing HCI servers.

In an HCI deployment, the storage and the compute functions run in the same nodes. Each server is simultaneously part of the storage system providing storage capacity, and a host providing compute resources. The challenges of this deployment mode are:

- **Harder to maintain** - Since compute maintenance impacts the storage and vice versa, careful planning needs to be carried out to ensure workloads are not disrupted while performing server restarts or updates to the cloud management platform, storage software, or operating system.
- **Higher workload density** - When node-down events occur due to hardware failures, HCI systems lose both compute and storage capacity, so proper planning is needed to ensure sufficient capacity to failover workloads and continue operations with minimal service disruptions.

- **Higher power density** - HCI systems are typically denser in terms of electricity requirements, so cloud builders need to ensure their data center partner can provision sufficient electric power per rack to support the HCI design.

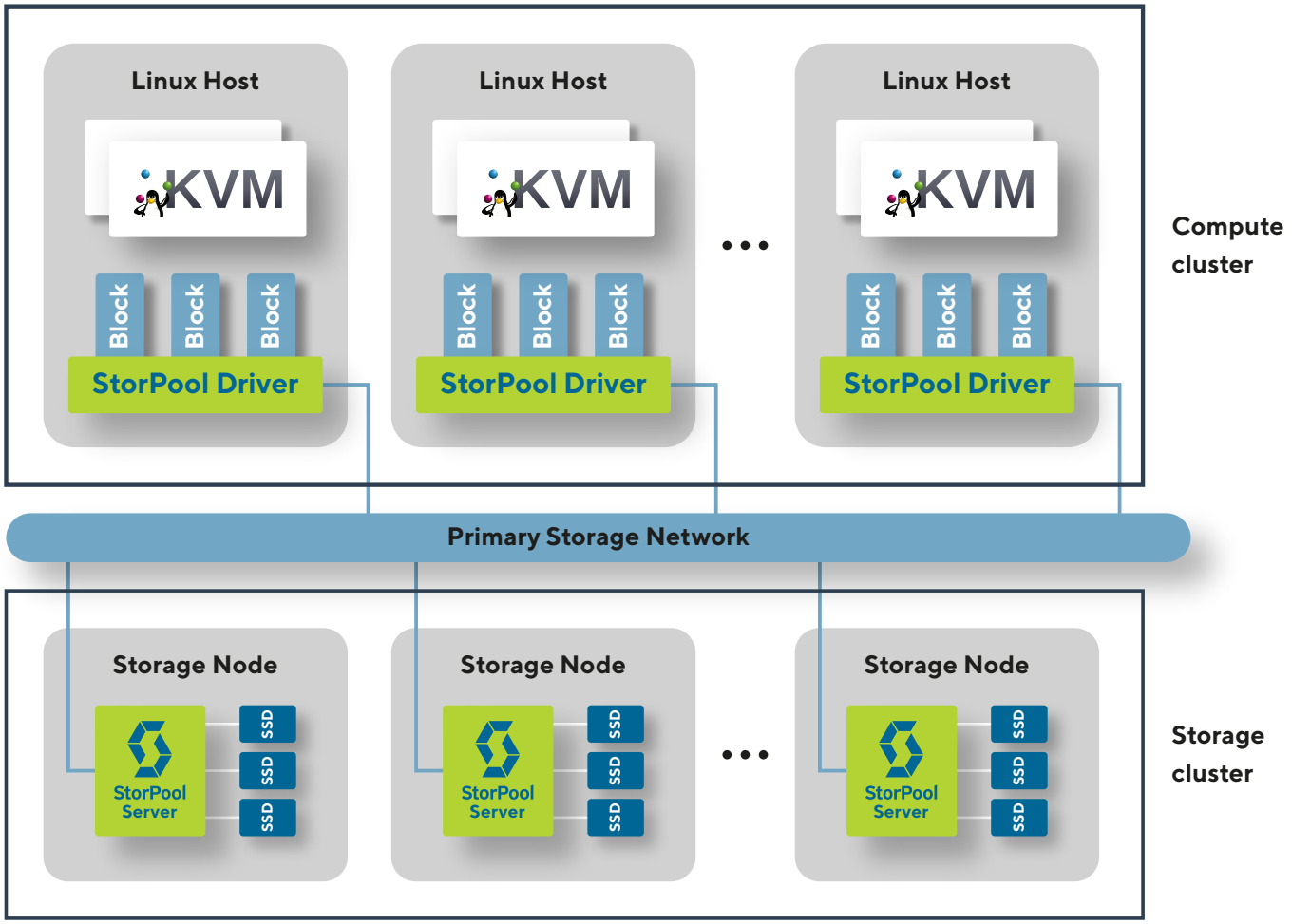
- **Meeting long-term infrastructure needs can be challenging** - In many cloud deployments, the capacity of one infrastructure component is used up faster than the other. Still, many HCI designs require that you scale with identical building blocks. With the Apache CloudStack and StorPool Storage solution, it is possible to start with an HCI design and later scale with independent hosts and storage nodes, so this challenge is diminished with proper planning.



In HCI deployments, the StorPool Storage software runs side-by-side with KVM virtual machines on the same servers. Although the storage devices are physically on the hosts, the data used by the virtual machines is distributed and replicated across all storage system nodes, just like when dedicated storage nodes are used. This ensures data protection in case of hardware failure and enables the scalability of the storage.

To prevent interference between different types of workloads, StorPool software is isolated by using dedicated CPU cores and pre-allocated memory. In a typical deployment, 2 to 4 CPU cores and 16-128GB of RAM are allocated to the StorPool software.

Disaggregated Infrastructure



Disaggregated deployments use dedicated servers for the storage system and the compute cluster. This type of deployment is preferred when purpose-optimized hardware will bring more value than a reduced total number of nodes. This type of deployment simplifies the independent scaling of storage and compute resources. The

benefits of the disaggregated cloud architecture are as follows:

- **Best value for money long-term** - The hardware resources for compute and storage servers can be streamlined from the start for the specific tasks each server needs to execute. As usage increases and the cloud scales, the total cost of growing and operating the cloud is optimized compared to using identical building blocks.
- **Easier to maintain** - With a disaggregated architecture, there is a clear delineation of who is responsible for which parts of the cloud infrastructure. In case external partners are hired to offload the tech team of the cloud builder, each company is responsible for their piece of the cloud and there is a low chance for cross-team interference. The same applies when different internal teams are responsible for each piece of the IT stack.
- **Higher availability of the whole cloud** - Since hosts and storage nodes can be maintained independently, the overall availability of the cloud increases slightly.
- **Easy to scale individual pieces** - Since the cloud infrastructure components can be added separately when needed, it is easy to scale the cloud as the need for more compute or storage resources arises.
- **Optimized storage node costs** - Given that StorPool Storage is very resource-efficient, storage nodes need a very small amount of resources - an 8-core CPU, 128 GB RAM, and 2 NIC ports are enough for a typical storage node. The disaggregated cloud architecture enables designing a cheap building block that is right-sized for storage.

Given that in a disaggregated deployment each server has a dedicated role and is a right-sized building block for the cloud, there are three main challenges of this deployment model:

- **Greater hardware variety** - it is more difficult to manage different hardware component availability for large-scale deployments
- **Limited ability to scale up** - in disaggregated clouds, scaling out is optimal for both compute and storage resources.
- **Larger physical footprint** - compared to HCI deployments, disaggregated clouds typically require more rack units to deliver the same amount of cloud resources.

High Availability

The presented solution implements high availability at multiple levels - Cloud management, hosts, storage, and network.

CloudStack Management Server

The CloudStack Management Server is deployed in a multi-node configuration. The management server itself is stateless and may be placed behind a load balancer. The state of the management server - cloud configuration, users, resources, etc are stored in a MySQL database. The availability and redundancy of the database are solved with two database servers with synchronous replication.

For small clouds, the CloudStack Management Server and the database can be deployed on a virtual machine with virtual disks on StorPool Storage. In this deployment type, the management of the VM and the HA is independent of the operation of the CloudStack Management Server. Availability and protection of the data are provided by StorPool Storage. In the event of a server failure, the VM is automatically restarted on another host.

Hosts

High availability is implemented through host redundancy. All hosts are active and run virtual machines. In case of a host failure, the VMs are automatically or manually restarted on the remaining hosts in the zone. This is possible because the hosts do not use local storage - all the data is on the shared primary storage - and any VM can run on any host in the zone. To allow this, the total available memory and CPU resources of the hosts are sized to handle all HA-enabled VMs even in case one host is offline.

Automatic VM failover is controlled by the CloudStack Management Server via its high-availability functions. This is enabled per VM. Virtual machines with HA disabled, will not be restarted automatically in the event of a host failure, but they can be restarted manually by an operator.

Primary Storage

The primary storage is the most critical component of the cloud in terms of availability. In StorPool Storage, data is protected via synchronous replication of each block of data to three storage nodes included in the storage system. This guarantees data availability in case of disk or node failures. Depending on the storage system size and architecture, the storage can tolerate a full rack outage without service interruptions or significant performance degradation.

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

Node failures are mitigated by implementing a shared-nothing distributed architecture. In case of a storage node failure, the remaining nodes continue to serve IO requests

from the hosts, and the system continues to work uninterrupted, without downtime or loss of data.

In case the storage system spans over multiple racks, it can be configured so that multiple copies of the data are stored in different racks. This setup tolerates a failure of a complete rack.

With end-to-end data integrity and self-healing, StorPool Storage 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.

Secondary storage

By offloading all the critical functions from the secondary to the primary storage, the secondary storage becomes a non-critical component in the cloud architecture. Failure of the secondary storage will result in an inability to create new templates, but will not affect other functions of the cloud. This tolerates short outages of the secondary storage.

In case cloud builders choose to simplify their cloud architecture by using the StorPool NFS service, the high availability of the secondary storage is ensured on two levels:

1. The NFS service is backed by a StorPool volume, so it benefits from the proven resilience of the StorPool block storage layer which distributes its data across all nodes in the StorPool storage system.

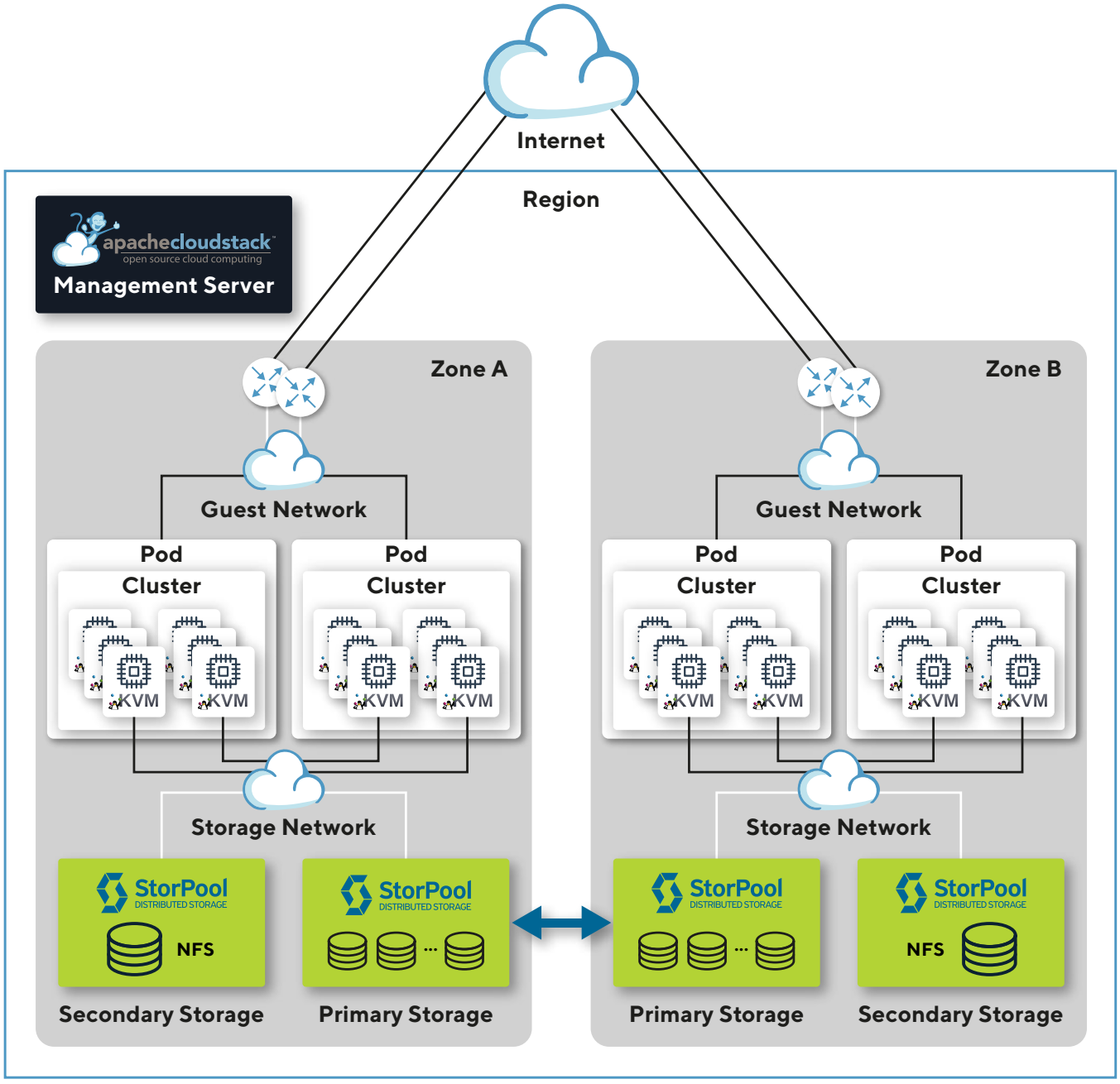
2. In node-down events, StorPool automatically switches over the NFS service to another node, ensuring the secondary storage availability.

Multi-Datacenter Deployments

The presented solution can be scaled to a multi-datacenter or to a multi-region cloud.

The multi-datacenter cloud contains multiple zones - one per data center, managed by one CloudStack Management Server. Each zone has local primary and secondary storage, guest network, and gateways. The multi-datacenter cloud shares a common user base, and the user can select the zone where the VM is deployed. The presented solution supports VM live migration within a zone, but not between zones.

A special feature of the multi-datacenter cloud with StorPool is the ability to make remote snapshots for backup or disaster recovery purposes. Depending on the needs, the remote data center may contain only a StorPool Storage system for the remote backups, or a full-blown CloudStack zone with hosts, primary and secondary storage, and a network.



Conclusion

The information provided in this document has been created from the collective information and experience of working with hundreds of cloud builders to help design and deploy hyperconverged and disaggregated cloud infrastructure. The document presents a high-level architecture and the pros and cons of the most popular deployment models for modern cloud deployments. Every cloud is unique but these common design principles should help readers plan how to build their cloud with Apache CloudStack and StorPool Storage.

This solution brief is designed to answer the main needs of cloud builders:

- Provide uninterrupted services
- Protect user data from loss in case of hardware failures and catastrophic events
- Ensure high and consistent performance adequate for demanding applications
- Simplify the daily operations tasks and eliminate maintenance downtime
- Scale easily with increasing demand

apachecloudstack
open source cloud computing

Interested to know more about Apache CloudStack?

Find Out More

StorPool
DISTRIBUTED STORAGE

Interested to know more about StorPool?

Find Out More