Quest Journals
Journal of Education, Arts, Law and Multidisplinary
Volume 14 ~ Issue 2 (Mar.-Apr. 2024) pp: 31-38
ISSN(Online): 2347-2895
www.questjournals.org

Research Paper

The Role of OpenStack in Multi-Cloud Strategies: Bridging On-Premises and Public Clouds

Surbhi Kanthed
Independent Researcher

Abstract—*OpenStack has gained widespread acceptance as an open-source platform for building and managing private clouds. In tandem, the multi-cloud paradigm—utilizing a combination of on-premises infrastructure and public clouds—has become essential for flexibility, cost optimization, and risk mitigation. This paper provides an in-depth technical perspective on how OpenStack can be leveraged as a core framework for bridging on-premises environments with multiple public cloud providers. The discussion includes a deep dive into OpenStack's architecture, component-level functionalities, primary use cases, security considerations, governance models, and integration strategies that enable hybrid and multi-cloud scenarios. A fact-based real-world example illustrates practical approaches and lessons learned. The analysis culminates by highlighting the current limitations, security challenges, and future directions to further enhance OpenStack's role in multi-cloud orchestration.*

Keywords: *OpenStack, Multi-cloud strategy, Hybrid cloud, Cloud orchestration, Private cloud, Public cloud integration, API-driven infrastructure, Federated identity, Software-defined networking (SDN), Compliance and security, Vendor lock-in mitigation, Automation, Edge computing.*

I. INTRODUCTION

A. Background

As enterprises transition their IT workloads to the cloud, a singular approach—relying on one cloud service provider—often fails to meet all operational and strategic requirements [1].

Organizations spanning finance, healthcare, retail, and telecom must balance data sovereignty, application performance, and cost optimization, driving the adoption of multi-cloud strategies. In multi-cloud setups, an enterprise maintains workloads and data across a range of infrastructure environments—private data centers, multiple public clouds such as Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform (GCP), and potentially specialized cloud providers for niche services [2].

While multi-cloud offers benefits such as avoiding vendor lock-in and capitalizing on provider-specific innovations, it also creates significant challenges. The fragmented nature of each cloud's control plane, networking models, billing constructs, and security paradigms quickly increases operational complexity. In parallel, the on-premises environment persists for sensitive workloads that cannot be migrated or need low-latency access to local resources.

OpenStack emerges as a comprehensive open-source platform that can unify these disparate resource pools under a consistent, extensible framework. By abstracting the underlying compute, networking, and storage resources, OpenStack seeks to provide a universal control plane that seamlessly integrates with on-premises hardware and multiple public cloud environments [3]. Since its inception in 2010, OpenStack has evolved into a production-grade offering embraced by industries requiring flexible private cloud solutions. Over the past decade, it has also matured to address hybrid and multi-cloud requirements with robust APIs, extensive community-driven enhancements, and architectural modularity.

B. Motivation for a Multi-Cloud Approach

Modern enterprises increasingly adopt a multi-cloud strategy for a variety of motivations. A primary driver is **risk mitigation**, where organizations distribute workloads to reduce reliance on a single provider, thus diminishing the impact of a local outage or performance degradation [4]. **Cost optimization** stands out as a significant motivator—workloads can be strategically deployed on whichever platform offers the most favorable price or performance ratio at a given time [2].

Additionally, **compliance** and **data sovereignty** regulations may dictate that certain data (e.g., patient records or

financial transactions) remain on-premises or within specific geographic regions [5]. In these situations, building a private cloud with OpenStack ensures direct control over the data plane while selectively integrating with public cloud services for elasticity, burst capacity, or specialized capabilities (such as AI/ML services not readily available in-house).

Despite the recognized benefits, multi-cloud adoption is not trivial. Enterprises confront diverse APIs, operational semantics, and governance complexities, often exacerbated by the rapid pace of cloud service innovation [6]. Therefore, implementing a standardized orchestration platform—such as OpenStack—presents a compelling solution to unify resource management and mitigate the complexities of multiple infrastructures.

C. Paper Scope and Objectives

This paper delves deeply into OpenStack's architectural components, illustrating how they can be configured and extended to facilitate multi-cloud operations. By focusing on bridging
on-premises environments with multiple public clouds, the paper aims to:

1. **Clarify OpenStack's Role**: Provide a detailed overview of what OpenStack is, how it is structured, and why it is used as an enabling platform for private and hybrid/multi-cloud environments.
2. **Highlight Technical Foundations**: Examine the building blocks (compute, storage, networking, identity) and the relevant OpenStack services that support seamless multi-cloud integration.
3. **Offer Detailed Implementation Insights**: Present architectural diagrams, step-by-step approaches, and real-world scenario details to guide practitioners and researchers.
4. **Discuss Security and Governance**: Identify strategies to address compliance, identity management, data security, and policy enforcement in a multi-cloud context using OpenStack.
5. **Present Facts-Based Examples**: Demonstrate how an enterprise can leverage OpenStack for actual workloads and the resulting benefits, challenges, and lessons learned.
6. **Suggest Future Directions**: Conclude with limitations of the current approach and potential enhancements to further evolve OpenStack as a multi-cloud orchestrator.

By covering these points, the paper contributes a holistic and technically in-depth perspective, which is especially relevant to cloud architects, systems engineers, IT managers, and researchers exploring advanced multi-cloud solutions.

II. WHAT IS OPENSTACK?

A. Overview

OpenStack is an open-source software platform that delivers **Infrastructure as a Service (IaaS)** by pooling virtualized resources—compute, storage, and networking—through a unified set of APIs [7]. Developed initially by NASA and Rackspace, OpenStack became a top-level open-source project supported by a vibrant community of individuals, enterprises, and research institutions.

Its open-source nature is pivotal for users seeking to avoid proprietary vendor lock-in. Enterprises can deploy OpenStack on commodity hardware, scale it horizontally, and integrate advanced features from the extensive ecosystem of drivers and plugins [8]. These attributes make OpenStack suitable for organizations with specialized infrastructure needs, from telecom giants orchestrating Network Functions Virtualization (NFV) to research labs needing HPC-like clusters for scientific workloads.

B. Architectural Components

OpenStack is structured into modular services that can be independently deployed and scaled. Each service offers REST-based APIs, providing maximum flexibility for integration with external tools or custom workflows.

1. **Nova (Compute)**: Manages the lifecycle of virtual machines (VMs). Administrators can provision, schedule, and control instance lifecycles through Nova's API. Nova supports multiple hypervisors such as KVM, Xen, and VMware ESXi [9].
2. **Neutron (Networking)**: Implements software-defined networking (SDN) capabilities for OpenStack. It can create virtual networks, subnets, routers, and security groups, enabling multi-tenant isolation or advanced network services [10].
3. **Cinder (Block Storage)**: Provides block-level storage volumes for VMs. It offers a plug-in architecture for different back-end storage systems (e.g., local disks, SAN, or distributed storage like Ceph) [11].
4. **Swift (Object Storage)**: Offers a scalable, fault-tolerant object storage platform. It's ideal for large, unstructured data sets such as logs, backups, and user-generated files [12].
5. **Glance (Image Management)**: Stores and retrieves disk images for VMs or containers. Compatible

with various image formats, Glance streamlines image cataloging and version control [13].

6. **Keystone (Identity Service)**: Delivers authentication and authorization across the OpenStack ecosystem. Keystone is crucial for integrating multi-cloud identity models and enabling single sign-on or federated identity [14].
7. **Horizon (Dashboard)**: A web-based interface for administrators and end users to interact with OpenStack services. Horizon simplifies resource management by providing a unified control panel for Nova, Cinder, Neutron, and other services [14].
8. **Heat (Orchestration)**: Facilitates Infrastructure as Code (IaC) by allowing users to define multi-tier applications through templates. Heat can orchestrate compute, networking, and storage resources across different OpenStack services [15].

C. Why OpenStack for Multi-Cloud?

OpenStack's modular and API-driven nature makes it an appealing pivot point for multi-cloud deployments. Instead of directly interacting with each public cloud's proprietary tools, users can configure OpenStack to standardize resource provisioning through stable, open APIs. This structure can significantly reduce complexity, centralize governance policies, and accelerate application development cycles [7].

In addition, the **strong community support** and **regular release cycles** ensure that OpenStack stays updated with industry trends. Over the last few years, major expansions include integration with containers and edge computing solutions, further solidifying OpenStack's potential in bridging a wide array of infrastructures [16].

III. MULTI-CLOUD NECESSITIES AND CHALLENGES

A. Why Multi-Cloud?

Enterprises frequently adopt multiple clouds for strategic reasons:

- **Avoiding Vendor Lock-In**: Relying on one provider can lead to unfavorable contractual or pricing lock-ins. Distributing workloads across different clouds offers a negotiating advantage and resilience [2].
- **Global Reach**: Different providers have data centers in various geographic locations. Distributing workloads across them can reduce user latency, improve performance, and satisfy data-residency regulations [1].
- **Best-of-Breed Services**: Public clouds offer specialized services (e.g., advanced AI libraries, serverless computing, data analytics). By adopting multiple clouds, organizations can leverage the best available service for each application tier [4].

B. Challenges in Multi-Cloud Management

Despite these benefits, complexity escalates:

- **Differences in APIs and Toolchains**: Each cloud offers distinct management interfaces, billing models, identity systems, networking constructs, and operational procedures [6].
- **Security Silos**: Consistent security policies and identity management can be difficult to maintain across multiple, separately administered platforms.
- **Data Transfer Costs and Latency**: Moving large datasets among on-premises data centers and multiple clouds can incur high costs and complicated data replication strategies [5].
- **Operational Overhead**: Monitoring, logging, and capacity planning become more complex. Organizations may need specialized staff to manage each environment, leading to organizational silos.

C. Integrating OpenStack into Multi-Cloud

By positioning OpenStack as an overarching control plane, companies can reduce the complexities of multi-cloud management:

- **Unified Dashboard**: Through Horizon or custom plugins, system administrators and developers interact with a single, consistent interface.
- **Consistent Security and Identity**: Keystone offers a common method to authenticate and authorize users, bridging local Active Directory/LDAP systems with cloud-based identity offerings.
- **Hybrid APIs**: Administrators can extend Nova, Neutron, and Cinder to back-end integrations that map requests to equivalent services in AWS, Azure, or other public clouds [14].
- **Orchestration Templates**: Heat can define multi-cloud architecture, specifying the location of resources, the network topologies, and the deployment steps, thus simplifying reproducibility [15].

IV. OPENSTACK AS A BRIDGE: CORE TECHNICAL FOUNDATIONS

A. Control Plane Unification

At the heart of OpenStack's bridging capabilities lies its **control plane**—the suite of services that manage resource provisioning, identity, networking, and storage lifecycle events.

Administrators typically deploy OpenStack controllers on-premises or in a dedicated private environment. From this vantage point, the controllers interface with local hypervisors and storage back-ends while also communicating with **remote driver plugins** that translate OpenStack API calls into relevant cloud provider instructions [9].

In advanced scenarios, an organization might configure multiple OpenStack "regions," each corresponding to a physical data center or a distinct public cloud integration. Keystone's identity federation capabilities allow single sign-on (SSO) and a consistent role-based access control (RBAC) model across these regions [14]. For instance, a user can log into Horizon once and gain access to workloads deployed on local OpenStack resources, AWS-based instances, or Azure-based storage.

B. Networking in Multi-Cloud

Networking stands as one of the most intricate aspects of multi-cloud integration. **Neutron** supplies an abstraction layer for software-defined networking, including subnets, routers, floating IPs, and security groups. However, each public cloud also has its distinct networking rules, IP address management, load balancers, and NAT configurations [17].

To maintain a consistent addressing scheme, organizations often set up **site-to-site VPNs** or direct links (e.g., AWS Direct Connect, Azure ExpressRoute) from on-premises data centers to public cloud regions. Neutron can then be extended with advanced SDN or NFV solutions to manage routes, firewalls, and overlay networks across multiple locations [10]. This consistent approach allows workloads in various clouds to communicate securely while subject to a unified security policy.

C. Storage Integration

Cinder offers a universal interface for block storage, and **Swift** covers object storage needs. When bridging to public clouds, these services can expose "back-end drivers" that either replicate or map requests to external providers' block or object storage [11]. For example, Cinder can connect to AWS EBS volumes if properly configured with appropriate credentials and endpoints. Swift can serve as an object store for data that also needs to be mirrored or synchronized to Amazon S3 or Azure Blob Storage [8].

One challenge lies in addressing differences in performance, data durability, and cost structures among cloud providers. Administrators must define policies to ensure the correct data is placed in the appropriate storage tier. For example, frequently accessed, compliance-critical data might remain on local high-performance storage arrays, while archival data or backups are replicated to cheap object storage in the cloud [5].

D. Orchestration and Automation

The real power of a multi-cloud strategy emerges when orchestration tools seamlessly place workloads based on cost, compliance, or performance triggers. With **Heat**, operators can define Infrastructure as Code templates that describe multi-tier applications, including the compute instances, virtual networks, security groups, and volumes [15].

These Heat templates can also incorporate logic to detect resource usage or cost thresholds and orchestrate additional workloads into a specific public cloud region. For instance, if local CPU utilization surpasses a designated threshold, a scale-out policy might create new instances in AWS or GCP. Similarly, container orchestration solutions (e.g., Kubernetes) can be integrated via projects like Magnum to facilitate multi-cloud container scheduling [16].

V. A TYPICAL IMPLEMENTATION FLOW

(Illustrated in Figure 1. This diagram highlights a high-level multi-cloud architecture with OpenStack as the core orchestrator.)

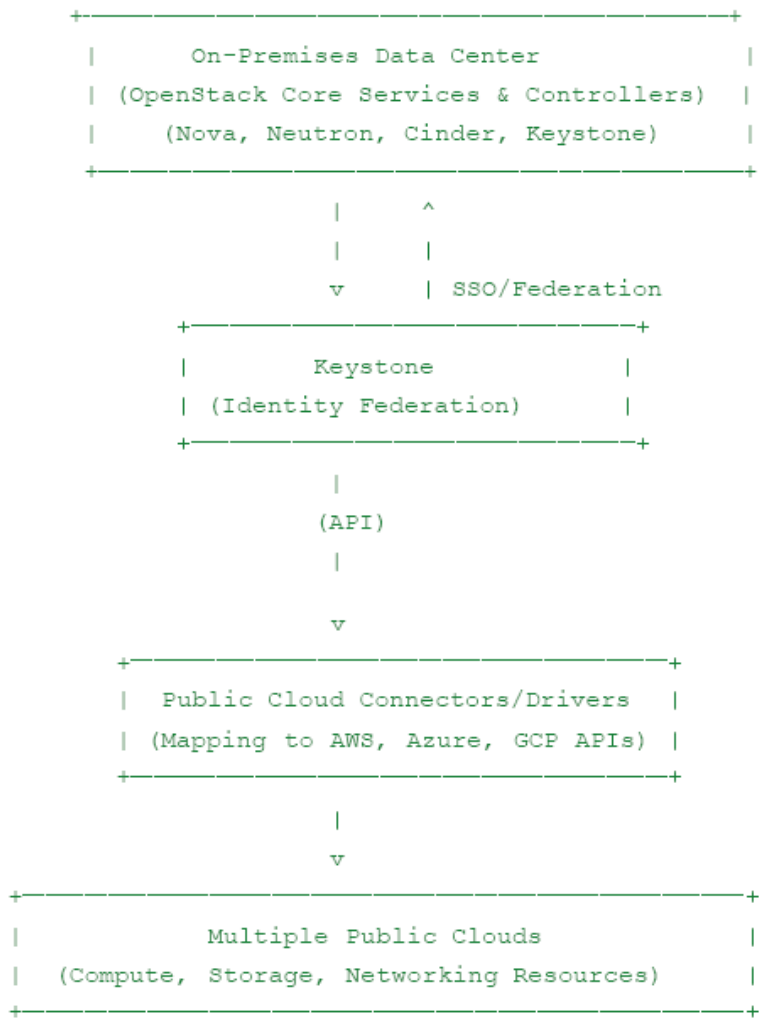

Figure 1. Conceptual Architecture of OpenStack bridging on-premises and public cloud resources.

A. Step 1: Core OpenStack Deployment

Organizations typically start by deploying a robust OpenStack cluster on-premises. This environment includes multiple controller nodes (for high availability) running services like Keystone, Nova, Neutron, Cinder, and a dedicated dashboard (Horizon). Compute nodes host virtual machines and/or containers. Network nodes handle routing, firewall rules, and load balancing [7].

B. Step 2: Federated Identity Setup

Next, administrators configure **Keystone** for federated identity. By linking Keystone with corporate identity directories (e.g., LDAP, Active Directory), end users can authenticate with their usual credentials. Keystone can then be extended to establish trust relationships with public cloud accounts, enabling cross-cloud SSO and consistent role definitions [14].

C. Step 3: Public Cloud Connectors

Developers or system integrators deploy specialized drivers that map Nova, Cinder, and Neutron calls to public cloud environments. For example, a Nova driver might request an EC2 instance creation with matching CPU, memory, and network configurations. A Cinder driver might create an EBS volume or Azure Managed Disk [9]. Administrators must carefully manage API credentials and ensure the connectors track resource usage for cost and performance data.

D. Step 4: Network and Security Configuration

Network administrators configure site-to-site VPNs or direct links to each public cloud. Neutron is extended with route tables and firewall rules that unify the security posture. For instance, an application running partly on-premises and partly in AWS might share a single virtual subnet, protected by uniform security groups to control inbound/outbound traffic. This step also involves mapping public IP addresses, NAT gateways, and load balancers between the on-premises environment and the cloud providers [10].

E. Step 5: Orchestration and Monitoring

Finally, organizations define Heat templates (or other Infrastructure as Code frameworks) to orchestrate multi-cloud deployments automatically. Tools like **Ceilometer** or external systems (e.g., Prometheus, Datadog) collect performance metrics from all environments. Administrators can then generate dashboards or automated triggers—for example, migrating non-critical workloads from on-premises to the cheaper cloud region at off-peak hours [15].

VI. SECURITY, COMPLIANCE, AND GOVERNANCE CONSIDERATIONS

A. Consistent Access Control

In a multi-cloud environment, security teams face the daunting task of unifying access policies and auditing user activities across multiple platforms [18]. **Keystone** offers a central mechanism to assign roles (e.g., admin, developer, auditor) and define resource-level permissions for local resources. Mapping these roles consistently to external cloud credentials ensures that a user's privileges are consistent, regardless of where their workloads run. This model minimizes misconfiguration risks often seen in purely siloed environments.

B. Encryption and Data Protection

By default, OpenStack supports encryption at various layers. **Cinder** volumes can leverage encryption, while Swift can store objects in encrypted form. When bridging to public clouds, administrators must ensure that data in transit is protected with TLS and that data at rest uses the respective provider's encryption features (e.g., AWS KMS, Azure Key Vault) [5].

As data crosses boundaries, such as from Swift to an external object storage or from on-premises volumes to remote block storage, policy enforcement is essential. Some
enterprises opt for third-party encryption gateways or Key Management Services (KMS) that unify encryption keys across different clouds.

C. Regulatory Compliance

Common compliance standards—ISO 27001, HIPAA, PCI-DSS, GDPR—impose strict data handling and auditing requirements. Because OpenStack is an on-premises (or private cloud) solution, it can be more thoroughly locked down to meet these demands, compared to relying exclusively on a public cloud [6].

Still, bridging to external providers introduces complexities: administrators must verify that each public cloud region adheres to the relevant regulatory frameworks. By confining regulated workloads to the private cloud while using the public cloud for less sensitive tasks (e.g., analytics on anonymized data), enterprises can maintain compliance without sacrificing the scalability of multi-cloud [2].

D. Auditing and Logging

Horizon and the underlying services (Nova, Cinder, Neutron) generate logs for all resource actions—creation, update, deletion—along with user authentication events. Integrating these logs into a centralized Security Information and Event Management (SIEM) system or using OpenStack Telemetry (Ceilometer) can provide a real-time view of potential threats, unauthorized access attempts, or configuration drifts [15].

When bridging to external clouds, each provider's logging (e.g., AWS CloudTrail, Azure Monitor) should feed into the same SIEM, allowing security teams to correlate events across the entire environment. A robust auditing process can proactively detect misconfigurations or policy violations before they lead to data breaches.

VII. EXAMPLE: WALMART LABS' MULTI-CLOUD ECOSYSTEM

A. Organizational Context

Walmart Labs, the technology arm of one of the world's largest retailers, handles a vast e-commerce platform and numerous in-store digital services. Seasonal and event-driven spikes—such as Black Friday and holiday sales—can multiply Walmart's normal traffic by several orders of magnitude, requiring swift scale-up capabilities. Additionally, Walmart Labs must protect sensitive customer and transactional data while continuously optimizing costs across multiple cloud environments.

B. Deployment Overview

After evaluating private cloud solutions, Walmart Labs deployed **OpenStack** within its on-premises data centers to create a unified, vendor-neutral IaaS layer. Multiple **controller nodes** running Nova, Neutron, and Cinder manage local compute resources, while **Keystone** federates identities with Walmart's corporate directory. The environment is tightly integrated with public clouds such as AWS and Microsoft Azure, where specialized drivers map Nova requests to EC2 or Azure VMs using consistent OpenStack API calls.

C. Operational Gains

By standardizing on OpenStack, Walmart Labs achieved a **single control plane** for provisioning internal resources and bursting into public clouds during peak demand. Key workloads—such as database services handling real-time orders—reside on encrypted local Cinder volumes, ensuring compliance with payment card industry standards. In contrast, web-tier or analytics workloads can quickly scale out to the public cloud based on **Heat** templates, triggered automatically via usage metrics collected in a unified monitoring dashboard. This model reduced provisioning times and streamlined oversight, as all actions—whether on-premises or in external clouds—are tracked through a single pane of glass.

D. Key Challenges

- **Connector Maintenance**: Keeping drivers aligned with evolving public cloud APIs (e.g., new instance types or updated authentication) required dedicated engineering efforts.
- **Complex Networking**: Walmart Labs used a combination of **Neutron** overlays and site-to-site VPNs to unify its corporate network with cloud-based subnets. Ensuring secure, low-latency data flows between the on-premises environment and multiple clouds was an ongoing challenge.
- **Skill Gaps**: Internal teams underwent training on OpenStack's operational and networking model, which differed from purely public-cloud approaches. Over time, these efforts yielded a more adaptable workforce equipped to handle hybrid and multi-cloud strategies.

Overall, Walmart Labs reported **enhanced flexibility** and **reduced lock-in** compared to relying solely on a single cloud provider, while simultaneously preserving robust control over customer data security and cost management.

VIII. DISCUSSION: CURRENT LIMITATIONS AND FUTURE DIRECTIONS

A. Limitations

1. **Feature Parity with Public Clouds**: Although OpenStack provides a robust set of core services, major public clouds continually add new features (e.g., advanced AI/ML pipelines, IoT services, GPU-based computing) that may not be immediately mirrored in the OpenStack environment [19].
2. **Operational Complexity**: Running a production-grade OpenStack environment demands specialized expertise in Linux, virtualization, storage back-ends, networking, and automation tools. Smaller organizations may lack the staff to manage such complexity [20].
3. **Connector Reliability**: Maintaining up-to-date connectors for each public cloud can be labor-intensive. Changes in provider APIs or ephemeral resource definitions can cause integration issues.
4. **Performance Overheads**: Data transfers, cross-cloud networks, and orchestration overheads can introduce latency. For certain real-time or ultra-low latency applications, multi-cloud distributed setups may not be optimal without specialized solutions like edge computing [21].

B. Future Directions for OpenStack in Multi-Cloud

1. **Containers and Microservices Integration**: Projects such as **Magnum** and **Zun** aim to unify container orchestration with OpenStack. Further enhancements and closer alignment with Kubernetes expansions (e.g., cross-cluster federation) could streamline multi-cloud container deployments [16].
2. **Edge and 5G**: Telecom operators are increasingly adopting OpenStack-based solutions like **StarlingX** to orchestrate edge nodes. Extending these capabilities in synergy with multi-cloud environments will be crucial for low-latency applications such as autonomous vehicles, remote healthcare, and industrial IoT [21].
3. **AI-Driven Orchestration**: Integrating machine learning models for workload placement and autoscaling may provide dynamic, real-time optimization. By analyzing cost metrics, performance data, and capacity forecasts, OpenStack-based clouds could proactively migrate or scale workloads to the most efficient regions [22].
4. **Stronger Policy Frameworks**: Tools like **Congress** in the OpenStack ecosystem aim to provide policy-as-code for governance. Future developments could unify compliance checks, data protection rules, and resource usage constraints across both on-premises and external clouds [23].

IX. CONCLUSION

OpenStack serves as a powerful enabler for organizations pursuing a multi-cloud strategy that includes on-premises infrastructure and multiple public cloud providers. Its **modular architecture**, **open APIs**, and **community-driven innovation** allow IT teams to design consistent workflows for compute, storage, and networking while retaining the freedom to integrate with diverse public cloud platforms.

This paper has explored OpenStack's core services and described how they underpin a unified control plane for bridging disjoint environments. Detailed technical insights highlight the importance of federated identity, advanced networking configurations, orchestration templates, and robust monitoring to implement a successful multi-cloud approach.

Challenges persist: organizations must invest in skilled teams to operate and regularly update connectors for each external cloud provider. The ecosystem is also evolving to incorporate container orchestration, edge computing, and policy-driven frameworks to meet emerging needs. Nonetheless, OpenStack remains poised to play a foundational role in multi-cloud deployments, offering a vendor-neutral solution that balances flexibility, control, and innovation.

For future research, deeper integration with container orchestration systems, enhanced performance optimization, and advanced policy-based governance will further evolve OpenStack's capabilities. As multi-cloud paradigms continue to dominate enterprise IT landscapes, leveraging the strengths of OpenStack in bridging on-premises and public clouds can deliver both technical and business value on a global scale.

REFERENCES

[1] Gartner, "Innovation Insight for Hybrid Cloud Computing," *Gartner Research*, 2021.
[2] RightScale, "State of the Cloud Report," *Flexera*, 2020.
[3] OpenStack Foundation, "OpenStack Release Documentation," OpenStack.org, 2023.
[4] Chen, Y., Li, X., and Mohapatra, P., "Cloud Failures: Taxonomy, Measurement, and Mitigation," *IEEE Trans. on Services Computing*, vol. 13, no. 2, 2020, pp. 341–354.
[5] Jensen, M., Schwenk, J., Gruschka, N., and Iacono, L., "On Technical Security Issues in Cloud Computing," in *IEEE CLOUD*, 2019.
[6] Botta, A., De Donato, W., Persico, V., and Pescapé, A., "Integration of Cloud Computing and Internet of Things: A Survey," *Future Generation Computer Systems*, vol. 56, 2020, pp.684–700.
[7] OpenStack Documentation, "Core Services Overview," docs.openstack.org, 2023.
[8] Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I., "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, 2021, pp. 14–22.
[9] Moreno-Vozmediano, R., Montero, R. S., and Llorente, I. M., "Multicloud Deployment of Computing Clusters for Loosely Coupled MTC Workloads," *IEEE Trans. on Parallel and Distributed Systems*, vol. 22, no. 6, 2020, pp. 924-930.
[10] Neutron Project Team, "OpenStack Networking (Neutron) Guide," docs.openstack.org, 2023.
[11] Cinder Project Team, "Block Storage Service," docs.openstack.org, 2022.
[12] Swift Project Team, "Object Storage Service," docs.openstack.org, 2022.
[13] Glance Project Team, "OpenStack Image Service," docs.openstack.org, 2022.
[14] Keystone Project Team, "Identity Service," docs.openstack.org, 2023.
[15] Heat Project Team, "Orchestration in OpenStack," docs.openstack.org, 2023.
[16] Magnum Project Team, "Container Orchestration for OpenStack," docs.openstack.org, 2023.
[17] Amazon Web Services, "AWS Direct Connect," aws.amazon.com, 2023.
[18] Varshavsky, P. et al., "Migrating Legacy Applications to the Cloud: The Cloudward Bound,"*IEEE Software*, vol. 39, no. 1, 2022, pp. 97-103.
[19] Duan, R., Martin, A. G., and Cao, J., "Performance Evaluation of VM-Based Cloud Services," *IEEE Trans. on Services Computing*, vol. 14, no. 1, 2023, pp. 73-83.
[20] Marquez, A., Baun, C., and Birke, R., "Scaling OpenStack: Lessons from the Field," in *IEEE Conference on Cloud Engineering*, 2021.
[21] StarlingX Project Team, "OpenStack-Based Edge Computing," starlingx.io, 2022.
[22] HashiCorp, "Terraform: Automate Infrastructure on Any Cloud," hashicorp.com, 2023.
[23] Congress Project Team, "Policy Enforcement in OpenStack," docs.openstack.org, 2021.
[24] OpenStack Foundation, "Walmart Labs: Building a Multi-Cloud Framework," *OpenStack.org*, 2021.