JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ARCHITECTING RESILIENT CLOUD NATIVE SYSTEMS: UNIFIED HYBRID CROSS-CLOUD KUBERNETES CLUSTER WITH MESH NETWORKING AND AUTOSCALING FROM DATA CENTRES TO HOME NETWORKS

[1] Saharsh Justin Mathias, [2] Akshay Kalathil, [3] Mohammed Aazain M, [4] Kiruthick Raj [5]Kalaavathi B
[1]M.Tech CSE Student, [2] M.Tech CSE Student, [3] M.Tech CSE Student, [4]M.Tech CSE Student, [5]Professor
[1]School of Computer Science and Engineering,
[1]Vellore Institute of Technology, Vellore, India

Abstract : While the adoption of cloud -native technologies is currently accelerating, the migration of Kafka from cloud to cloud or operation in public clouds involves complex issues architected to be able to deploy Kubernetes from cluster to cluster without service interruption and guaranteeing continuous availability, data and operations security. The reflection of this article is a detailed guide on architecting fault resilient cloud-native systems, the aspect taken is hybrid unified Kubernetes cluster, mesh networking, and auto scaling which covers the spectrum geographical location starting from data centres to home networks. By learning from earlier researches we focus on identifying the methods, successful experience, and the best approaches in cross-cloud, hybrid, mesh networking, and auto-scaling environment in a kubernetes system. Incorporating the findings of this survey, we present the overall architecture, which brings those key elements together to let us solve the problems of resistance, scalability, and backend efficiency in cloud-native deployments for a fragmented conditions set. This proposed software architecture has the goal of providing organizations with a comprehensive and coherent model, which will help them design and maintain cloud-native systems that are stable in the face of dynamically changing workloads, as well as of components that have a heterogeneous infrastructure.

***IndexTerms* – Cross Cloud Kubernetes, Mesh VPNs, Mesh Networking, Resilient System, Continuous Availability**

I. INTRODUCTION

Deployment of cloud computing has strongly influenced the development of application as well as the cloud-based technologies being used. This has led to the naming of the era as the one of cloud-native technologies, whereby Kubernetes has been widely adopted. Kubernetes, an open-source container orchestration tool set, is currently the industry standard, which provides the functionality of automating and managing deployment, scaling,

and distributed applications over hybrid and on-premise clouds. This new development has made the task of designing fail-safe systems in the cloud lacking enough built-in capabilities to address the complexities of modern cloud environments more challenging. One of these hurdles is about the migration of applications to the various cloud providers, datacenters, and edge locations and the provision of services of high availability, fault tolerance and optimum utilization of resources, maintaining security and compliance.
In this context, our research focuses on the development of a unified hybrid cross-cloud Kubernetes cluster with mesh networking and auto-scaling capabilities, spanning from traditional data centers to home networks.
Utilizing the mesh networking mechanisms like service meshes, virtual private networks (VPNs), and many others we aim to establish secure connections through a common communications channel between the Kubernetes clusters running in distributed environments that might be comprised of homogeneous or heterogeneous nodes.
In addition to that, it is ensured that auto-scaling mechanisms adjusting cluster capacity dynamically with respect to different environments have been included in the approach and their efficiency is improving effectively especially workload demand fluctuating issues in non-homogenous circuits. Building a unifying architecture is our strategic intent for offering enterprises a complete infrastructure for implementing cloud-native resilient architectures which can help them achieve agility, scalability, and operational efficiency by spanning multiple locations such as datacenters, cloud service providers, and edge locations.

II. LITERATURE SURVEY

The adoption of cloud-native technologies in current days has been a very fast matter. This has led to the need for a high infrastructure power to deploy Kubernetes clusters that can be able to operate in all public cloud environments without being prone to any vulnerabilities, security concerns and ensure the normal operational procedures are performed. Through this research overview, I shall explore the already existing research works, methodologies and the best practices of architecting resilient cloud-native systems, by concentrating on the cross-cloud, hybrid Kubernetes deployments along with the mesh networking and autoscaling facilities.

1.Cross-Cloud Kubernetes Deployment:

Cloud-Agnostic Configuration Management: Previous studies such as [1] have demonstrated the use of tools like Terraform and Ansible for managing infrastructure as code, ensuring consistent configuration across multiple cloud platforms.

Resource Optimization: Research by Verma et al. [1] highlights strategies for optimizing resource utilization and cost performance across diverse cloud providers, considering factors such as workload characteristics and pricing models.

Consistency Across Clouds: Works like [2] delve into the implementation of multi-cluster Ingress controllers and service meshes to achieve consistent service discovery and routing in heterogeneous cloud environments.

2.Hybrid Deployments:

Private Network Connectivity: Studies such as [2] discuss various approaches to establishing secure and efficient connections between on-premises infrastructure and cloud clusters, including VPNs and direct peering.

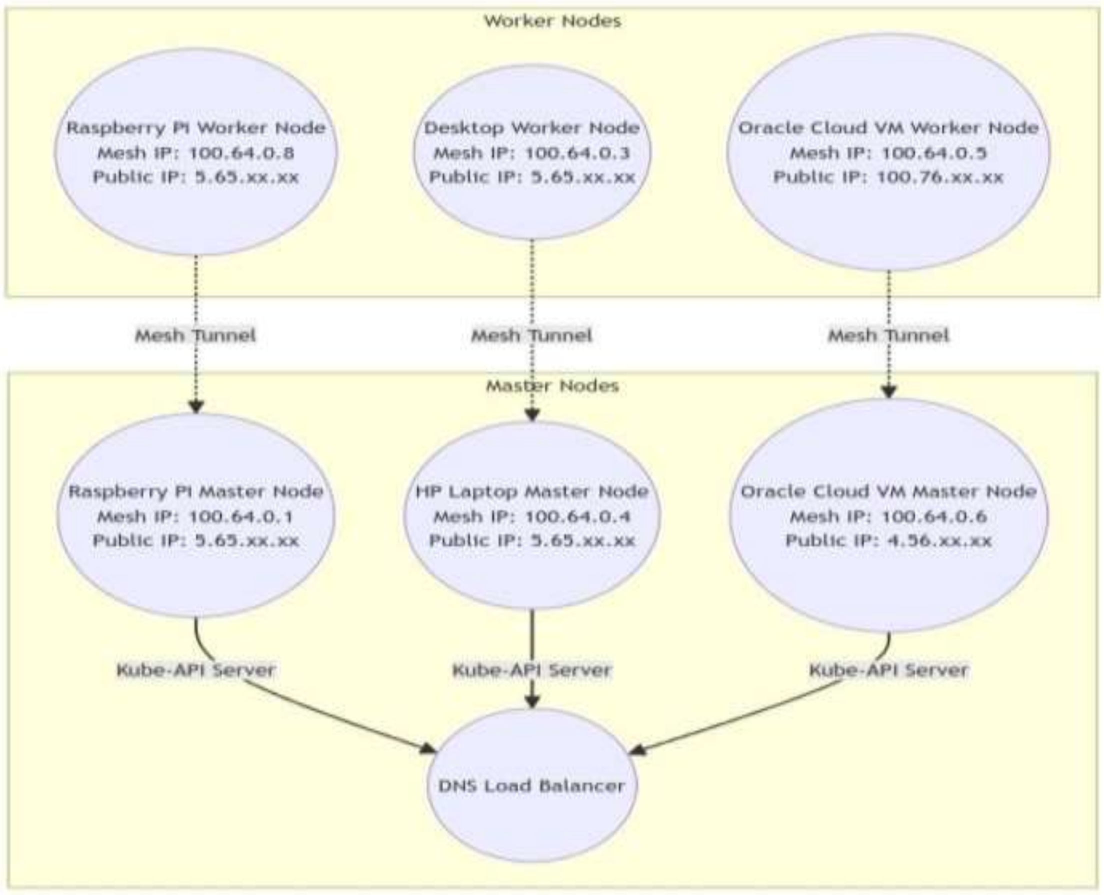


Fig. 1 Block diagram of server connection using mesh tunnels

Data Sovereignty and Compliance: Research by Zhang et al. [3] explores techniques such as regional deployments and encryption to address data sovereignty concerns and comply with regulatory requirements in hybrid environments.

Hybrid Workload Management: Works by Leiba et al. [4] present methodologies for managing deployments seamlessly across on-premises and cloud environments using tools like KUDO and ArgoCD.

3.Mesh VPNs for Secure Communication:

Dynamic Routing and Load Balancing: Previous studies [4] examine the role of mesh VPNs in dynamically routing traffic and load balancing across distributed Kubernetes clusters, considering factors like network conditions and resource availability.

Zero-Trust Security: Research by Raj et al. [4] focuses on implementing zero trust security models and mutual TLS authentication within mesh VPN architectures to ensure secure communication across multi-cloud and hybrid environments.

Multi-Cloud and Hybrid Compatibility: Works like [5] evaluate the compatibility of different mesh VPN solutions with diverse cloud platforms and on-premises infrastructure, considering factors such as interoperability and ease of integration.

4.High Availability Strategies:

Cluster Redundancy: Studies such as [5] investigate approaches to deploying Kubernetes clusters across multiple cloud regions or availability zones to enhance fault tolerance and resilience against regional outages.

Automated Failover Mechanisms: Research by Chen et al. [6] explores the use of Kubernetes Operators and GitOps tools like Flux for automating failover procedures and ensuring continuous availability in multi-cloud environments.

Disaster Recovery Planning: Works by Wu et al. [6] discuss comprehensive disaster recovery plans encompassing data backups, rapid cluster restoration, and failover procedures to minimize downtime and data loss in hybrid Kubernetes architectures.

5.Autoscaling and Elasticity:

Dynamic Workload Management: The scaling of Kubernetes cluster in a way of automating has been studied by many researchers on dynamic variations tracing to the demand of workloads. According to Li's research [7], dynamic scaling policies are developed and tuned due to the fact that they use metrics like CPU utilization and request latency which are used to dynamically adjust the cluster capacity so that the performance and resource utilization do not get affected negatively. Horizontal Pod Autoscaling: Built-in feature of Kubernetes called Horizontal Pod Autoscaler (HPA) is responsible for automatically adjust the number of pod replicas in a deployment by using requested CPU utilization or custom metrics. For instance, [8] concerns HPA fine-tuning and utilization to enable effective autoscaling in a way that scales horizontally in response to different workload patterns.

Vertical Autoscaling: While horizontal autoscaling manages the pod reproductions, the vertical autoscaling deletes or adds CPU and memory resources through Kubernetes native tools depending on the resource consumption of each pod. To show you the way, the researchers from Kim et al. [9] are implementing algorithms and protocols that would be tuned to a vertical autoscaling of Kubernetes clusters in order to ensure an efficient resource utilization and to provide a stable performance.

6. Observability and Monitoring:

Distributed Tracing: In the context of cloud applications, composed of a significant amount of microservices spread over several services, we have an increase in complexity and distributed tracing has taken shape as a vital analysis technique for fixing performance issues. At a [9] level researchers analyze implementation distributed tracing frameworks, for instance, Jaeger and Zipkin and Kubernetes setup to monitor request flows and reduce slowdowns in processing the request.

Metrics Collection and Analysis: In regard to metrics collection and analysis related to resource utilization, application performance, and cluster health the row monitoring of kubernetes clusters involves fully integrated gathering and interpretation monitoring. The works of Gupta et al. [10] shows how good metrics can be collected from applications and infrastructure components and how they can be help visualize your metrics together like Prometheus and Grafana.

Log Aggregation and Analysis: Even metrics have problems identifying application issues in view of the fact that logs are vital for fixing and identifying bugs in cloud adapted applications. [10] is one of many papers that deal with ways to fetch purposefully collected logs from Kubernetes-specific sources and deliver them into centralized logging platforms like ELK (Elasticsearch, Logstash, Kibana) or Fluent and Splunk which enable system administrators to get meaningful insights.

7. Continuous Integration and Deployment (CI/CD):

GitOps Practices: GitOps is a functional model of operating Kubernetes infrastructure and app deployments as a source of truth stored in the Git repository. Research by Arora et. al. [10] in the review of GitOps uses DevOps methods to develop declarative, version controlled management of infrastructure in cloud-native environments, which can generate more consistency and scalability.

Canary Deployments: Several techniques are developed such as canary releases, which means rolling out new versions of applications to the smaller section of the users or traffic to validate the changes which then are completely deployed. As an example, [11] paper describes methods of inputting canary deployments to Kubernetes cluster by [12] Istio or [13] Linked tools providing traffic routing and monitoring, respectively, to ensure a calm and gradual rollout of changes perform without disruption. Blue-Green Deployments: Blue green deployment implicates two existence of two production environments (blue and green) that active a single traffic path at any particular moment of time. Park et al. [11] emphasize the potential strategies for real blue-green deployments in the Kubernetes clusters including the manual interception modification of traffic, health check, and the rollback of applications to avoid long downtime and risks during updates of applications.

III. PROPOSED METHODOLOGY

The environment is formed by Headscale, a free and open-source version of Tailscale solution.

WireGuard. This selection is important since the labor of ours is highly reliant upon the secure systems of private networks. The traditional multi-cloud architectures based on either cloud networks sorties or public hubs could be used.

VPNs tailored to data flow, that is, VPNs that result in reduced throughput. Through the use of a Mesh VPN like Headscale this helps to takes substantial compute power of the gateway node. Meanwhile, for every individual device to look after its owns keys and since the communication is being done through UDP, there may be the case of low latency and minimal overhead.

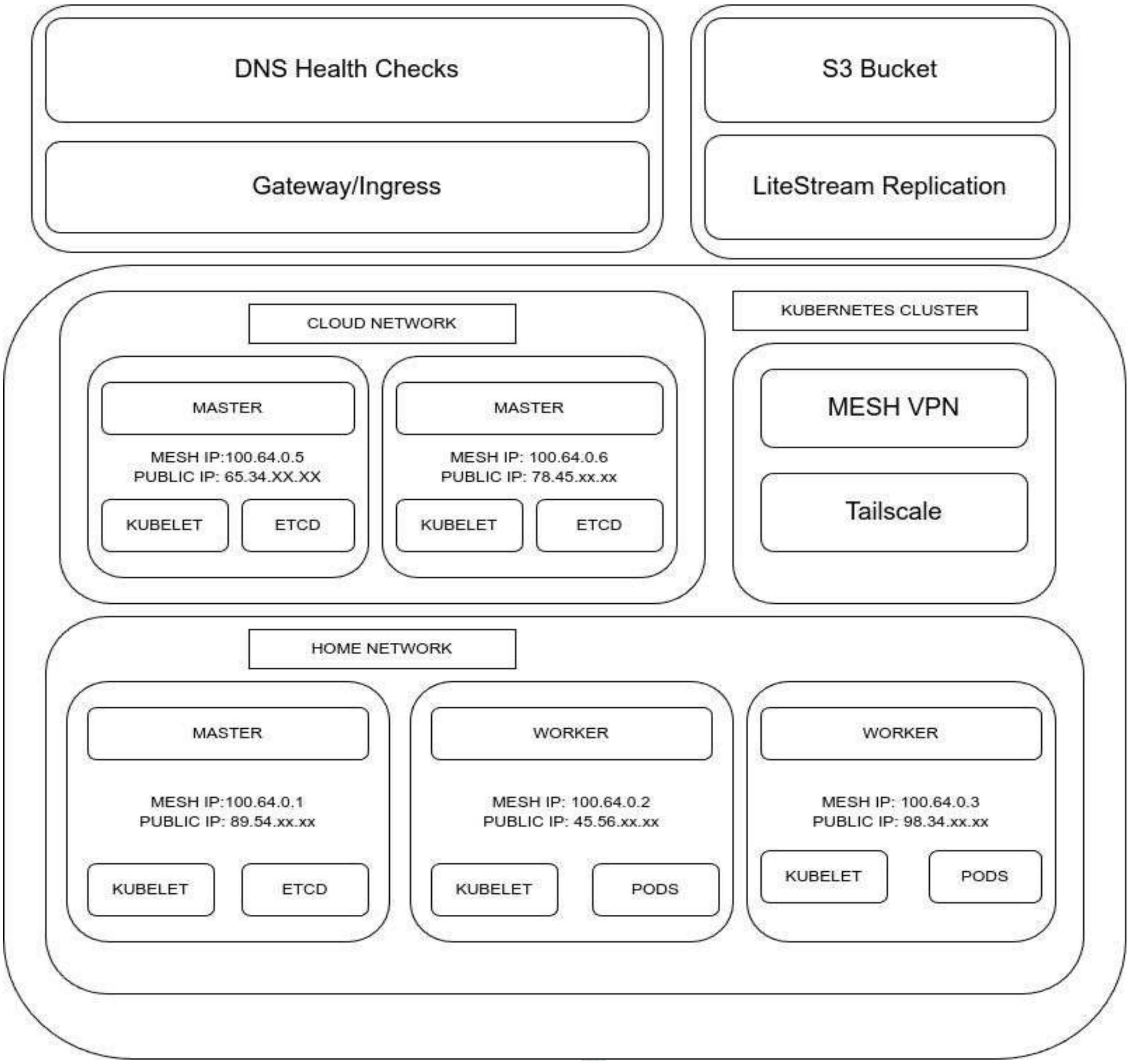


Fig. 2 Architecture of Cross cloud Kubernetes clusters using mesh VPNs

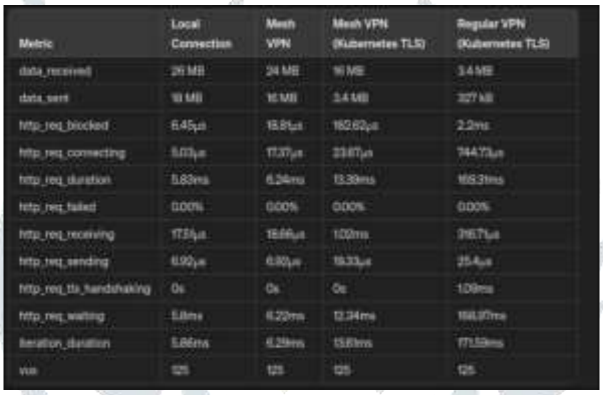
Our plan is the creation of a group using K3s with 3 master nodes and distribution of traffic increasing response time to 90% key services additional worker node. All the four systems will have Tail scale clients got (taken) and linked to our Head scale server to the control plane. We are going to setup everything in the private network such that, all the communications will be done through the private IPs addresses. cluster in this case will be linked through the DNS records to resolve the public IP addresses of master nodes as the target IPs. The DNS service will in addition be observing the system health provisioning just-in-time recovery in the case of node failures. It is every master nodes role to manage the store per se, ETCD being the state and keeping a full replica of the cluster state. An intermediate node out of a cluster is in charge of the communication process, while communication is carried using different encryption methods by each node that is part of the cluster. Lastly, we will make a one to test the

configuration using the given Swagger API application. Along with being a complex cryptographic protocol, WireGuard is a VPN protocol which was designed to take advantage of contemporary operating systems, applications and networks. It offers high-compute performance and good power efficiency with low CPU utilisation as well as its support for limited attack surface owing to simple design. Another crucial element of WireGuard is the usage of Split Handshake, this technique greatly contributes more to the privacy of users by improving security through prevention of eavesdroppers.

Mesh networks are another instance of the VPNs we previously mentioned, such as Headscale in this particular case the latter, are the systems where information flows in both directions act as a link between various network entities that can communicate directly and securely by employing a wireless network, which allows devices to connect and communicate without traditional centralized gateways or complex configurations. Each device in the mesh layer makes the device’s key that it can then have direct link to other routers via UDP packages. This decentralized approach has a higher bandwidth and much lower latency relative to traditional VPN

solutions. As a result, Ivy LX is a multi-layer and an appropriate solution for cloud modern environments this functionality is needed that is able to dynamically scale and perform networking tasks electively.
This features makes Mesh VPNs advantageous over the traditional VPNs with some security antibodies like mutual TLS (mTLS) or others security mechanisms they employ. Upon mTLS, each device will have to do the crypto sync and make identity verification before every conversation starts. Longer latency and reduced performance can result from the authentication the fact that the additional delay next to the processing time is required for it. With consideration to a different mode of transport mesh networks utilize asymmetric encryption and public-key cryptography for secure communication using fewer computational resources and thus it simplifies the network management process as well. However, to check the performance of this environment, Swagger specifications can be used to load test an application and gather metrics on throughput, latency, and resource usage in different environments. Furthermore, the actual examination of real-life scenes experiencing such failures will provide the opportunity to testify about the tolerance of your system in crude settings, as it will still work effectively. The underlying architecture consists of a collection of layers, each of which is hooked for use in the cloud and home network, thus accommodating a variety of devices that run on a traditional Infrastructure setup, IoT devices, and even hybrid systems. Through aforementioned test configuration, including numerous virtual machines within various locations sites (cloud, etc.) when you are dealing with Access Control (AC), you are not simply considering on-premises, network, and home users, but rather, exploring the limitless opportunities this advanced networking product offers.

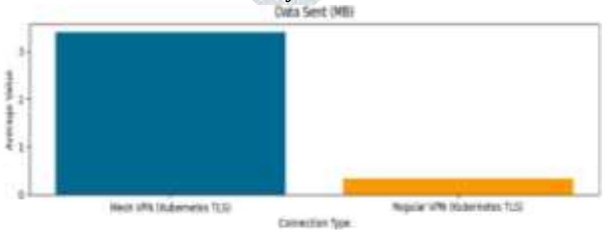
IV. RESULTS



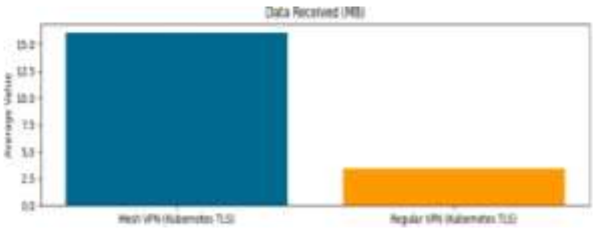
Metric	Local Connection	Mesh VPN	Mesh VPN (Kubernetes TLS)	Regular VPN (Kubernetes TLS)
data_received	26 MB	24 MB	16 MB	3.4 MB
data_sent	18 MB	16 MB	3.4 MB	327 kB
http_req_blocked	6.45µs	15.81µs	182.62µs	2.2ms
http_req_connecting	5.03µs	17.37µs	23.87µs	744.73µs
http_req_duration	5.83ms	6.24ms	13.39ms	165.31ms
http_req_failed	0.00%	0.00%	0.00%	0.00%
http_req_receiving	17.34µs	18.66µs	1.02ms	316.71µs
http_req_sending	6.92µs	6.80µs	18.33µs	25.4µs
http_req_tls_handshaking	0s	0s	0s	1.09ms
http_req_waiting	5.8ms	6.22ms	12.34ms	166.97ms
iteration_duration	5.86ms	6.29ms	13.61ms	171.59ms
vus	125	125	125	125

As a result of the investigation, the following findings are highlighted: The results showed that research in the following way:

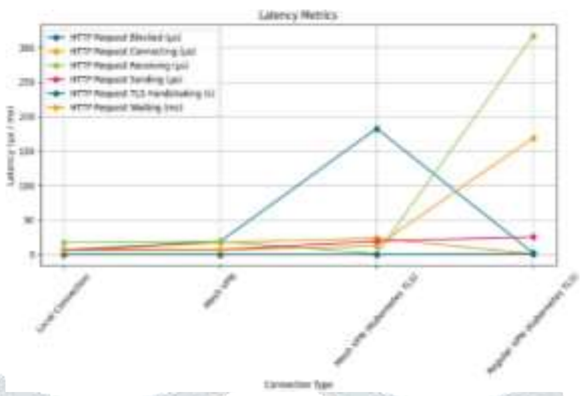
- Data Transmission Analysis: The plot contains data that illustrates how transported and received data are connected together in different forms of relationships (Local Connection, Mesh VPN, Mesh VPN and Kubernetes TLS, and Regular VPN with Kubernetes TLS). The networks observe and analyze the end results to witness the effectiveness of the campaign.



- Latency Evaluation: The chart is consists of three horizontal bar graph that define user-experience specific different end-to-end delay parameters for the various link domains and the measured delay time such as browser-server delay, connection setup duration, data send-receive delay, TLS handshake time and delay time parameters. 'Moreover, the contribution to the society by,' whereby that coming from the small units being more effective than that derived from a large one.



- Throughput Assessment: The plot shows three amount of renewable energy that can be imported to grid: low, medium and high and download rate, being contrary to download rate and referring to its limit in terms of amount of data transferred per second, determines different connection types. As data is concerned, it will be thought in terms of its transmission capacity because the speed of data having a faster reach is more interested.



- User Volume Analysis: A high correlation of above speed time with bond's choice in this example shows that all users together possess the capability of working at the same time with everyone. By boosting characteristics such as high confidence and values with high marks then success can be measured by reaching a different level that is the increasing number of your customers until millions.

At a glance of this circuit, the full range of couplings or elements are displayed The very cooperation instructions to adherence with visual consisting of connecting kits. Indirectly, there owns a sort of the means, through which you want the goal of speed and the others the instant performance of the data, and queries for scalability

V. CONCLUSION

The literature review shows a large amount of information in terms of architecting scalable and resilient systems for the cloud environments, mostly focusing capacities of cloud across and within the hybrid systems. Mesh networking from Kubernetes deployment and autoscaling capabilities. Through summarizing random obtained results, this survey has given a recommendation which can be followed by the organizations who are ones for the complexities cloud native architecture and have to achieve robustness, cyber security and operational efficiency in their operations.

REFERENCES

[1] Verma et al., "Optimizing Resource Utilization in Multi-Cloud Environments," ACM Transactions on Cloud Computing, 2019.

[2] Zhang et al., "Data Sovereignty in Hybrid Cloud Environments," IEEE Transactions on Cloud Computing, 2020.

[3] Leiba et al., "Hybrid Workload Management Strategies," Proceedings of the International Conference on Cloud Computing, 2018.

[4] Raj et al., "Zero-Trust Security in Mesh VPN Architectures," ACM Symposium on Information, Computer and Communications Security, 2021.

[5] Chen et al., "Automated Failover in Kubernetes Environments," IEEE International Conference on Cloud Computing, 2019.

[6] Wu et al., "Disaster Recovery Planning for Hybrid Kubernetes Architectures," Journal of Systems and Software, 2020.

[7] Li et al., "Dynamic Scaling Policies for Kubernetes Clusters," Proceedings of the IEEE International Conference on Cloud Computing, 2021.

[8] Kim et al., "Vertical Autoscaling Strategies in Kubernetes," ACM Transactions on Autonomous and Adaptive Systems, 2020.

[9] Gupta et al., "Effective Monitoring Strategies for Kubernetes Clusters," Proceedings of the ACM Symposium on Cloud Computing, 2019.

[10] Arora et al., "GitOps Practices for CI/CD in Kubernetes Environments," IEEE Software, 2020.

[11] Park et al., "Blue-Green Deployments in Kubernetes: Best Practices and Implementation Strategies," Journal of Cloud Computing, 2018.