**SIFY’s Kubernetes Service**

Service Guide V1.1

Prepared by Swaminathan Subramanian

Table of Contents

[1. Business Background 2](#_Toc159338427)

[2. Key features and benefits 3](#_Toc159338428)

[3. Service Description 5](#_Toc159338429)

[3.1 Kubernetes Service Description 5](#_Toc159338430)

[3.2 Service Variants 6](#_Toc159338431)

[3.3 Service Product Code & Catalog 7](#_Toc159338432)

[3.4 Service Region and Locations 7](#_Toc159338433)

[3.5 Service deliverables 7](#_Toc159338434)

[4 Sify Kubernetes Service Architecture 9](#_Toc159338435)

[4.1 How it works 11](#_Toc159338436)

[4.2 Kubernetes Cluster Access Methods 12](#_Toc159338437)

# Business Background

In the rapidly evolving landscape of modern business and technology, Cloud-Native Applications have emerged as a pivotal innovation in the Software development and application release management. Organizations are compelled to adopt digital technologies to innovate, improve operational efficiency, meet evolving customer expectations, and create new revenue opportunities. Digital transformation allows businesses to harness data, enhance agility, and respond swiftly to market changes, ultimately ensuring their long-term viability and growth in the digital era.

In the verge of rapidly evolving technological landscape, businesses face increasing demands for agility, scalability, and reliability in managing their containerized applications. Kubernetes, an open-source container orchestration platform, has emerged as the industry standard for container management and orchestration. As organizations embrace cloud-native applications and microservices architectures, the need for a comprehensive Kubernetes solution becomes more evident.

Sify Technologies stands out as a premier Managed Service Provider offering a cutting-edge Multi-Cloud Kubernetes Platform tailored for Cloud-Native Applications and their mobility requirements. This platform boasts distinctive features and delivers compelling value propositions, including direct network connectivity with high-speed bandwidth across top-tier cloud providers like Oracle Cloud, AWS, Azure, and Google Cloud.

# Key features and benefits

**K8S** **Cluster Provisioning, Automation & Orchestration**

* Simplified consistent provisioning interface for Kubernetes across Private & Public Cloud
* Automate the process of scaling, upgrading and decommission Kubernetes Clusters
* Unified dashboard to deploy, manage and monitor kubernetes cluster across different environments.

**Multi-Kubernetes-Cluster Management**

* Federation of Kubernetes clusters to manage and synchronize resources like deployments, services, and ingress objects.
* Simplifies management of DNS across multiple kubernetes clusters by leveraging Global DNS
* Simplified web-interface to deploy applications, manage resources, and monitor performance.

**Hybrid and Multi-Cloud Enablement**

* Direct Cloud Connect facilitates hybrid and multi-cloud strategies by seamlessly integrating on-premises infrastructure with various cloud environments.
* Enables flexibility in workload placement, disaster recovery, and application mobility.

**Cluster Security, Compliance and Policy Management**

* Fine grained access control for Cluster, workloads with customized roles and permissions
* Multi-Tenancy of kubernetes clusters, namespaces, and projects to isolate & manage across environments.
* Simplify the process of defining and managing kubernetes network policies, to secure network traffic between pods and namespaces.

**Low Latency and High-Speed Cloud Connect**

* Cloud Connect minimizes network latency and guarantees high-speed data transfer between an organization's on-premises data centers or office locations and hyper-scale cloud environments.
* This low-latency connectivity is crucial for real-time applications, data-intensive workloads, and seamless user experiences.

**Container Security & Identity**

* Container Runtime Security, Vulnerability Scanning, Network Segmentation and Firewall
* Automated compliance checks, CSI benchmarks, Shift left methodology with CI/CD Integration
* Quarantine affected containers, deep packet inspection, real time visibility into container security & events.

**Traffic Management, Monitoring & Observability**

* Advanced traffic Management, Ingress Management, load balancing , Service Mesh Integration
* Built-in Monitoring, real time metrics, alerts and notifications, Logging and Auditing, Distributed tracing
* Dashboard Customization, ITSM integration, and Email based alerting.

**Optimized Cloud Services**

* With Sify DC Cloud connections to hyper-scale cloud providers, enterprises can optimize the performance of cloud-based applications, services, and data storage, ensuring efficient utilization of cloud resources.

**Application Mobility**

* Sify's kubernetes solution empowers organizations with the ability to move microservices across different cloud providers with ease.
* This enhances workload flexibility, facilitates disaster recovery strategies, and minimizes vendor lock-in.

**Disaster Recovery and Business Continuity**

* The platform's application mobility capabilities facilitate robust disaster recovery and business continuity strategies, ensuring minimal downtime and data loss of Microservices workload deployed on Sify’s Kubernetes Landscape

**Vendor Neutrality**

* Sify's platform promotes vendor neutrality, allowing you to leverage the strengths of different cloud providers without being locked into a single ecosystem.

# Service Description

# Kubernetes Service Description

Sify’s Kubernetes as a Service is a cutting-edge solution designed to empower businesses with seamless container orchestration and management. Leveraging the power of containerization technology, our Kubernetes Service offering enables organizations to build, deploy, and scale applications efficiently, driving agility and innovation in the rapidly evolving digital landscape. The service layer basically abstracts the underlying infrastructure complexities, allowing developers to focus on building and deploying applications without the need to manage the infrastructure where the containers run.

Below are the services offered under Sify’s Kubernetes as a Service as part of its service delivery -

* **Container Orchestration**

Kubernetes Service platforms typically include a Kubernetes orchestration layer that automates the deployment, scaling, and management of containers. This ensures that applications run consistently and reliably across different environments.

* **Container Registry**

Kubernetes Service often provide a container registry to store and manage container images. This registry facilitates the versioning, distribution, and sharing of container images within an organization.

* **Infrastructure Abstraction**

Kubernetes Service abstracts the underlying infrastructure, such as servers and networking, providing a simplified interface for developers and operators. This abstraction enhances portability, allowing applications to run seamlessly across various cloud providers or on-premises environments.

* **Resource Scaling**

Kubernetes Service platforms enable dynamic scaling of containerized applications based on demand. This can be done manually or automatically, allowing organizations to efficiently allocate resources and handle varying workloads.

* **DevOps Integration**

Kubernetes Service supports DevOps practices by providing tools and APIs for integration into continuous integration/continuous deployment (CI/CD) pipelines. This ensures a seamless and automated process for building, testing, and deploying containerized applications.

* **Service Discovery**

Kubernetes Service platforms often include service discovery mechanisms, making it easier for containers to find and communicate with each other. This is crucial for microservices architectures, where multiple services need to work together.

* **Security Features**

Kubernetes Service services implement security features such as container isolation, role-based access control (RBAC), and secure communication between containers. These measures help protect applications and data running in containers.

* **Logging and Monitoring**

Kubernetes Service platforms typically provide tools for logging and monitoring containerized applications. This includes collecting and analysing logs, as well as monitoring performance metrics to ensure the health and performance of the deployed containers.

* **Networking Solutions**

Kubernetes Service offer networking solutions that facilitate communication between containers and external services. This includes features such as load balancing, network segmentation, and support for isolation of Clusters.

* **Multi-Tenancy**

Kubernetes Service supports multi-tenancy, allowing multiple users or teams within an organization to deploy and manage their containerized applications independently, while sharing underlying compute resources.

* **API Access**

Kubernetes Service platforms provide APIs that allow developers and operators to programmatically interact with and manage containerized applications. This enables automation and integration with other tools and systems.

# Service Variants

|  |  |  |
| --- | --- | --- |
| **Kubernetes Service Variants** | **Standard** | **Enterprise** |
| **Cluster Access** | CLI | CLI, UI |
| **Access Control** | Config File | RBAC |
| **Dashboard Access** | NA | Unified Dashboard |
| **Application Catalog** | Helm Chart | Pre-Built App Catalog |
| **Multi-Cluster Management** | NA | Unified Cluster Dashboard |
| **Monitoring and Logging** | Third-Party Integrations | In-Built Monitoring Dashboard |
| **Availability SLA\*** | Standard availability 99% | High Availability 99.9% |
| **Backup & Restore** | Manual Third-Party Integration | In-built Catalog |
| **Persistent Volume** | Manual Third-Party Integration | In-built Distributed Block Storage |

NOTE: \* Based on high availability architectural design of the cluster

# Service Product Code & Catalog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Product** | Sub product | Plan | Description | Description |
| CI-PAAS | OK8S-app | OK8S-app-per-MKN | Opensource K8S - Master Node - K8S Latest version - Dedicated K8S- One Node per cluster | Opensource Kubernetes per one master node |
| CI-PAAS | OK8S-app | OK8S-app-PER-WKN | Opensource K8S - Worker Node - K8S Latest version - Dedicated K8S- One Node per cluster | Opensource Kubernetes per Worker node |
| CI-PAAS | ENTK8S-app | ENTK8S-app-PER-MKN | Enterprise K8S - Master Node - License Per Node - Dedicated K8s - One Node per cluster | Enterprise Kubernetes per one master node |
| CI-PAAS | ENTK8S-app | ENTK8S-app-PER-WKN | Enterprise K8S - Worker Node - License Per Node - Dedicated K8s - One Node per cluster | Enterprise Kubernetes per Worker Node |
|  |  |  |  |  |
| CI-PAAS | K8S-LB | K8S-LBS-PRIVATE | Software Load Balancer - HA Proxy - Dedicated Private LB - 2 HA Proxy Nodes | Kubernetes software load balancer with HA. Dedicated two nodes |
| CI-PAAS | K8S-LB | K8S-LBS-PUBLIC | Software Load Balancer - HA Proxy - Dedicated Public LB - 2 HA Proxy Nodes - 1 Public IP | Kubernetes software load balancer with HA. Dedicated two nodes. Mandatory to purchase one Public IP. |

# Service Region and Locations

Sify’s Kubernetes Services are offered out of two Cloud enabled datacentre locations in India. Mumbai and Bangalore are the two current regions or service locations in India.

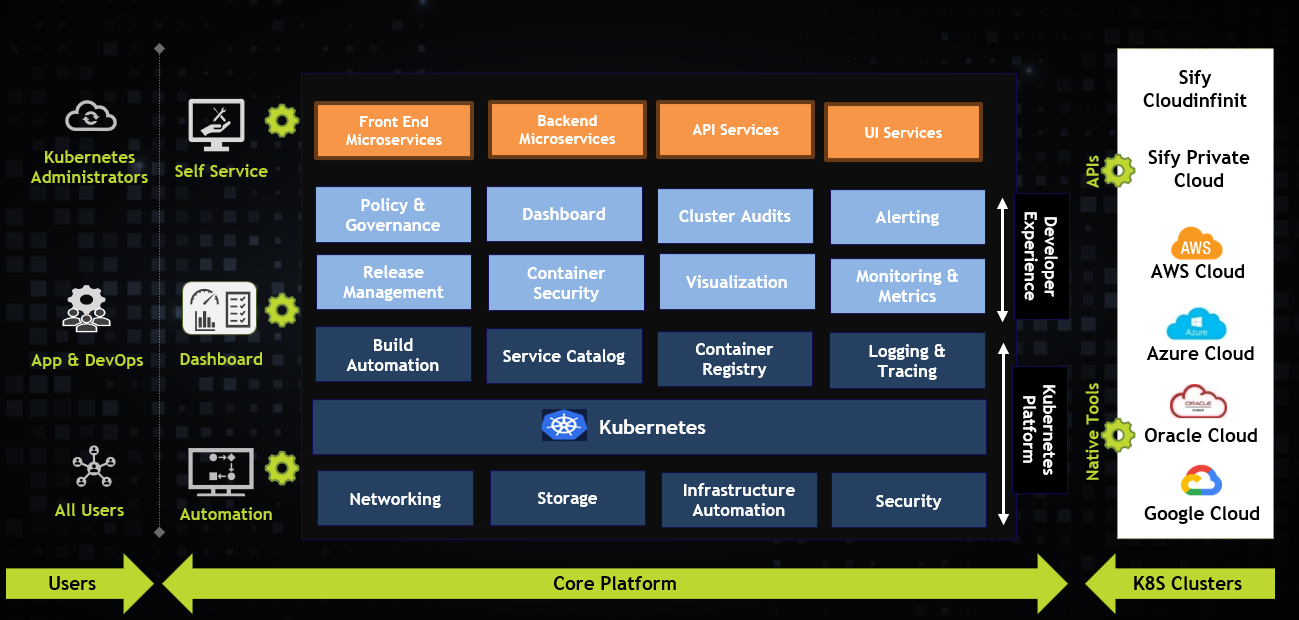
# Service deliverables

Sify Kubernetes as a Service is a managed platform that abstracts the complexities of Kubernetes infrastructure and operations for customers. Below are the key deliverables that Sify Kubernetes as a Service (KaaS) offers:

|  |  |
| --- | --- |
| **Key Deliverable** | **Description** |
| **Cluster Provisioning and Management** | Automated provisioning and management of Kubernetes clusters, including the control plane and worker nodes. |
| **High Availability (HA) Configurations** | Configuration options for setting up high-availability clusters to ensure redundancy and minimize downtime. |
| **Infrastructure Integration** | Seamless integration with cloud provider services, such as load balancers, storage solutions, networking components, and identity and access management (IAM) services. |
| **Node Management** | Adding or removing worker nodes based on resource requirements and demand. Scaling node pools to accommodate varying workloads. |
| **Security Features** | Built-in security measures, including identity and access controls and network policies to enhance the security posture of the Kubernetes clusters. |
| **Application Deployment and Scaling** | Assisting with the deployment of applications using Kubernetes manifests or Helm charts. |
| **Monitoring and Logging** | Integrated monitoring and logging solutions for tracking the health, performance, and logs of Kubernetes clusters. |
| **Automated Scaling** | Tools and features for automatic scaling of the clusters based on resource utilization or custom-defined metrics. |
| **Identity and Access Management (IAM) Integration** | Integration with IAM systems for managing access controls and permissions for customers, ensuring secure and controlled access to the Kubernetes clusters. |
| **Infrastructure as Code (IaC) Support** | Support for defining and managing Kubernetes clusters using Infrastructure as Code (IaC) tools, allowing customers to version-control and automate their configurations. |
| **API Access** | Access to the Kubernetes API, allowing customers to interact with the cluster programmatically and integrate with their tools and automation workflows. |
| **Backup and Disaster Recovery** | Tools and processes for backing up and restoring cluster configurations and data to facilitate disaster recovery and cluster migration. |
| **User Interface (UI) for Management** | A user-friendly web-based interface for managing and monitoring Kubernetes clusters. This can include features for deploying applications, managing resources, and monitoring cluster health. |
| **Resource Scaling and Optimization** | Tools for monitoring resource utilization and optimizing cluster resources to ensure efficient use of compute, memory, and storage. |
| **CI/CD Integration** | Integration with CI/CD pipelines to facilitate automated deployment, scaling, and updates of applications running on Kubernetes clusters. |
| **Troubleshooting and Issue Resolution** | Investigating and resolving issues related to cluster components, networking, and applications. Providing support for diagnosing and troubleshooting application-specific problems. |
| **Upgrades and Patch Management** | Automated upgrades for Kubernetes versions and patch management for security updates to keep the clusters up to date. |
| **24X7 management and Customer support** | Sify’s qualified, experienced professionals manage and support the Kubernetes platform on 24 X 7 basis. |
| **Service Level Agreements (SLAs)** | Clear SLAs outlining the level of service and support provided by the KaaS provider, including uptime guarantees, response times, and resolution times for issues. |
| **Documentation and Support** | Comprehensive documentation and support resources to assist customers in understanding the platform features, troubleshooting issues, and optimizing their use of Kubernetes. |

Above deliverables collectively create a managed Kubernetes service that enables customers to focus on deploying and managing their applications, leaving the operational aspects of Kubernetes infrastructure to the service provider. Moreover, it's crucial to ensure a reliable, secure, and user-friendly experience to customers.

# Sify Kubernetes Service Architecture



**Kubernetes Components**

**Master Node**

The Master Node is the command centre of a Kubernetes cluster, responsible for orchestrating and managing the entire system. It houses critical components that collectively form the control plane, making global decisions about the cluster and ensuring its stability and functionality.

**API Server**

* Acts as the gateway to the Kubernetes control plane.
* Exposes the Kubernetes API, receiving and processing user and application requests.

**etcd**

* A distributed key-value store that stores the configuration data and state of the cluster.
* Serves as the single source of truth for the cluster's information.

**Controller Manager**

* Runs controller processes that monitor the state of the cluster.
* Takes corrective actions to ensure the desired state is maintained.

**Scheduler**

* Assigns pods to worker nodes based on factors like resource requirements and constraints.
* Ensures optimal distribution of workloads across the cluster.

The Master Node is crucial for managing the overall state of the cluster and making decisions to maintain the desired state. It abstracts complexity, providing a centralized interface for users to interact with and control the Kubernetes environment.

**Worker Node**

The Worker Node is where the actual workloads run—the place where containers execute and applications are deployed. It collaborates with the Master Node to fulfil the cluster's operational requirements.

**Kubelet**

* An agent that runs on each Worker Node.
* Ensures containers are running within pods and communicates with the Master Node.

**Kube-proxy**

* Maintains network rules on nodes, enabling communication to and from pods.
* Facilitates services like load balancing.

**Container Runtime**

* Executes containers within pods.
* Supports various runtimes like Docker, containerd, and others.

The Worker Node is the execution engine, responsible for running and managing containers. It communicates with the Master Node to receive instructions, report the status of workloads, and contribute to the overall functioning of the cluster.

**Kubernetes Dashboard**

The Kubernetes Dashboard is a web-based user interface that provides a graphical representation and management console for Kubernetes clusters. It offers an intuitive way to interact with and manage resources within a Kubernetes environment. The Kubernetes Dashboard is not a mandatory component of Kubernetes, but it is a popular and useful tool for cluster administrators, developers, and users who prefer a graphical interface for certain tasks.

In summary, the Master Node is the brains behind the Kubernetes cluster, making decisions and managing the control plane. The Worker Node is the brawn, executing workloads and ensuring applications run seamlessly. Together, they form a robust and scalable architecture for deploying and managing containerized applications in Kubernetes.

**Software Load Balancer**

HAProxy, which stands for High Availability Proxy, is an open-source software load balancer and proxy server that is commonly used in Kubernetes environments to distribute network or application traffic across multiple backend servers. It plays a crucial role in ensuring high availability, scalability, and reliability for applications deployed in Kubernetes clusters. HAProxy in Kubernetes, it is often configured as part of the Ingress resource or as an external load balancer.

**Bastion Node**

In the context of Kubernetes or other distributed systems, a bastion node (also known as a jump host or jump server) is a specialized server that serves as an entry point into a secure network. It acts as an intermediary or gateway through which users can access other servers or resources within the network. The primary purpose of a bastion node is to enhance security by controlling and monitoring access to sensitive systems.

# How it works

Sify’s Kubernetes as a Service is designed to run on any linux based operating system deployment on HyperScale cloud or On-Premises. As part of the offerings Sify Kubernetes as Service is offered in Two different variants as Standard and Enterprise.

Sify’s Kubernetes as a Service also provides you the full access and ability to manage the control plane nodes that are responsible for scheduling the containers, managing application availability, storing cluster data and other key tasks. Sify Kubernetes as Service platform is designed on upstream kubernetes which allows the developers / users to run the existing plugins and tools from existing Kubernetes community. Applications running on Sify Kubernetes platform are fully compatible with applications running on any standard kubernetes environment, whether running in on-premises data centers or public clouds. This means that one can easily migrate the standard kubernetes applications to Sify Kubernetes platform without refactoring the code. Any existing applications that are running on Cri-O & Containerd as a runtime are compatible to migrate and run in Sify Kubernetes Platform.

Sify Kubernetes as a Service lets you run your Kubernetes applications on Sify’s Virtual Private Instances (VPI) and Virtual Private Enterprise (VPE) Compute Services. With Sify’s VPE and VPI Compute Services, you can take advantage of all the performance, scale, reliability, and availability of required compute infrastructure, as well as integrations with Sify cloud networking and security services, such as load balancer for load distribution support for pod networking.

Customer can subscribe to the Sify Kubernetes Service based on the compute requirements like number of Master Nodes, number of worker nodes and the amount of CPU, RAM and Storage required. A Minimum of 1 Master Node and 2 Worker Nodes is recommended for Non-Production Environments. For Production environments, minimum of 3 Master Nodes and 3 Worker Nodes is recommended to ensure High Availability of the Environment. The compute resources required can be subscribed and the resource can be vertically scaled on demand at the Virtual Instance level. In terms of storage requirements, the kubernetes nodes can be integrated with Block, File or Object Storage by subscribing from the Sify Storage Service Catalog.

**Sify Kubernetes as a Service Infrastructure Deployment Features**

|  |  |
| --- | --- |
| **Deployment Feature** | **Description** |
| **Provisioning** | Sify Kubernetes as a Service Deployment requires provisioning of below cloud resources to run and support containerized applications:   * Sify Cloud Compute (VPE / VPI) – both for Master and Worker Nodes. * Sify Cloud Storage – for each Virtual Machine. * Sify Kubernetes platform supports Ubuntu, CentOS and RedHat Linux. * Load Balancers, if needed. * Other Security and Network Components such as Firewall, NAT Gateway. |
| **Customization** | Sify Kubernetes as a Service Deployment also provides flexibility to choose custom sized resource configurations for Master and Worker nodes in Dedicated Environment to run and support containerized applications seamlessly. |
| **Scalability** | Sify Kubernetes as a Service offers both Vertical and Horizontal scaling of Pods as per the need of Application availability and to meet changing demands. |
| **Unified Dashboard** | Sify Kubernetes as a Service Deployment provides a unified dashboard to view and manage the Kubernetes platform deployed across Sify Cloud and other Hyperscale cloud – AWS/Azure/ GCP/ OCI. |
| **Monitoring & Logging** | Sify Kubernetes as a Service Enterprise edition has in-built monitoring and logging solution which collects & records metrics for monitoring & alerting. Also the logging mechanisms allows for storing of logs for analysis & compliance purpose. |

# Kubernetes Cluster Access Methods

**Sify Enterprise Kubernetes Web-Based Interface**

Description: Sify Enterprise Kubernetes offers a web-based graphical user interface that allows users to manage and interact with their Kubernetes clusters. The UI provides a user-friendly experience for cluster administration, application deployment, and monitoring. This enterprise interface allows for managing multiple Kubernetes cluster across On-Premises and Public Cloud Kubernetes Platform

Use Case: Ideal for users who prefer a visual interface for managing and monitoring their clusters.

**Sify Standard Kubernetes Dashboard**

Description: Kubernetes Dashboard is a web-based user interface that provides a visual representation of a Kubernetes cluster. It allows users to view and manage cluster resources through a graphical interface. This is a default Kubernetes dashboard which comes as a service within the kubernetes cluster with minimal features as part of the dashboard.

Use Case: Useful for users who prefer a graphical interface for monitoring and managing their clusters.

**Kubectl (Kubernetes CLI)**

Description: kubectl is the standard command-line tool for interacting with Kubernetes clusters. Sify Kubernetes Platform provides a convenient option to download a kubeconfig file directly from the Sify Cloudinfinit dashboard, allowing users to use kubectl for cluster operations.

Use Case: Developers and administrators often use kubectl for tasks such as deploying applications, managing pods, and inspecting cluster resources.

**API Access**

Description: Kubernetes Cluster platform provides a comprehensive RESTful API that allows users to programmatically interact with Kubernetes clusters. The API supports a wide range of operations for cluster management.

Use Case: Developers and automation workflows can use the API to integrate kubernetes functionality into custom applications, scripts, or other tools.

**Integration with GitOps Tools**

Description: Integrates with GitOps tools like Flux and Argo CD, allowing users to manage and deploy applications on Kubernetes clusters through a Git repository. Changes in the repository trigger updates to the clusters.

Use Case: GitOps practitioners can leverage these integrations for automated and version-controlled application deployments.

**kubeconfig File**

Description: The kubeconfig file contains cluster information, user authentication details, and context settings. Users can use the kubeconfig file to configure kubectl and other Kubernetes tools to connect to a specific cluster.

Use Case: Configuring kubeconfig allows users to switch between different clusters and manage authentication details.

**SSH Access to Nodes**

Description: Users can SSH into individual nodes in the cluster to perform tasks, inspect logs, or troubleshoot issues directly on the nodes.

Use Case: Advanced troubleshooting or debugging scenarios where direct access to node-level information is necessary.

**Helm (Package Manager for Kubernetes)**

Description: Helm is a package manager for Kubernetes that simplifies the deployment and management of applications. Users can use Helm charts to define, install, and upgrade applications on Kubernetes clusters.

Use Case: Streamlining the deployment of complex applications with dependencies.

**Kubernetes Managed Service RACI MATRIX**

|  |  |  |
| --- | --- | --- |
| **Activity** | **Sify** | **Customer APP Team/ Developer** |
| **Responsible for deploying and managing K8s Infrastructure. Responsible for enforcing security and compliance** | **Application Developers and DevOps team responsible for deploying and managing applications** |
| **Resource Sizing - Compute, Storage** | C, I | R, A |
| **Infrastructure Provisioning** | R, A | C, I |
| **IAM Policies** | R, A | C, I |
| **Cluster Deployment** | R, A | C, I |
| **Network Policy Configuration** | R, A | C, I |
| **Application Deployment** | I | R, A, C |
| **Define required Security Controls** | C, I | R, A |
| **Security Controls Implementation** | R, A | C, I |
| **High Availability & DR - Infrastructure** | R, A | C, I |
| **High Availability & DR - Application** | C, I | R, A |
| **Monitoring & Logging** | R, A | C, I |
| **Ongoing maintenance of new cluster deployments and existing cluster lifecycle management** | R, A | C, I |