# National Scholarship Portal

## National Informatics Centre | Govt. of India

## New Delhi

**The National Scholarships Portal (NSP)** is an initiative introduced by the Government of India to simplify and improve the scholarship application and disbursement process. It serves as a centralized digital platform where students from all across the country can apply for a wide range of centrally sponsored scholarships offered by various government departments, ministries, and state governments. The primary objectives of the NSP are to enhance transparency, efficiency, and accessibility within the scholarship process and provide financial aid to deserving students pursuing higher education in India.

**Key Features and Benefits of the NSP:**

1. **Centralized Application Process**: NSP offers a single platform where students can apply for multiple scholarships provided by different government bodies. This streamlines the application process and eliminates the need for students to navigate multiple websites or physical paperwork.

2. **Multiple Scholarship Schemes**: The portal encompasses various scholarship schemes, including pre-matric, post-matric, and merit-based scholarships, catering to students from diverse academic backgrounds and educational levels. This broad coverage ensures that a wide range of students can benefit from the portal.

3. **Transparency and Accountability**: NSP ensures transparency by providing real-time updates on the status of scholarship applications, disbursement details, and scholarship amounts. Students can track their applications and receive timely notifications about their application status.

4. **Timely Disbursement**: Through a streamlined online application process, NSP aims to expedite the disbursal of scholarship funds directly into the bank accounts of eligible students. This reduces delays and minimizes paperwork, ensuring that financial assistance reaches students when they need it the most.

5. **User-Friendly Interface**: The portal is designed with a user-friendly interface to make it easy for students to navigate, register, and apply for scholarships without encountering technical difficulties. This design encourages more students, including those in rural areas, to access and utilize the platform.

6. **Mobile Application**: NSP offers a mobile application, enabling students to access the portal and apply for scholarships using their smartphones. This mobile app extends accessibility, especially in remote or underserved areas where internet access via smartphones may be more prevalent than traditional computers.

7. **Dashboards for Authorities**: In addition to serving students, NSP provides dedicated dashboards for educational institutions and state authorities. These dashboards facilitate the management of scholarship applications, document verification, and the disbursement process, ensuring efficient administration.

8. **Digitization of Documents**: The portal allows students to upload and submit their required documents online. This digitization of documents reduces the need for physical paperwork, making the application process more environmentally friendly and efficient.

## Problem areas / Pain points

- Manual war deployments are often slow and infrequent due to time-consuming manual testing and deployment processes.
- In larger development teams, integrating code changes from different developers can be challenging, leading to integration conflicts.
- In case of issues with a release, rollbacks can be challenging and time-consuming.
- Developers may face issues when deploying code to different environments due to inconsistencies in configuration.
- Monitoring required to identify the problem or service status and resolve it quickly.
- Customers were not getting real-time insights into the health and performance of systems and applications, so mostly they received problems facing mail through users.
- Problem solving was taking time because manually checking the cause of the problem will take time.
- Users require monitoring the resources allocated across different components of the setup. This information helps in optimising resource distribution and avoiding resource conflicts.
- Users want to handle the scaling and load balancing of containers and ensure high availability of applications.
- User wants pod scaling management itself as per the application load.
- Haproxy  setup required with Custom rule in configuration for managing high load of users.

## Problem Statement:

1. Manual war deployments are often slow and infrequent due to time-consuming manual testing and deployment processes.

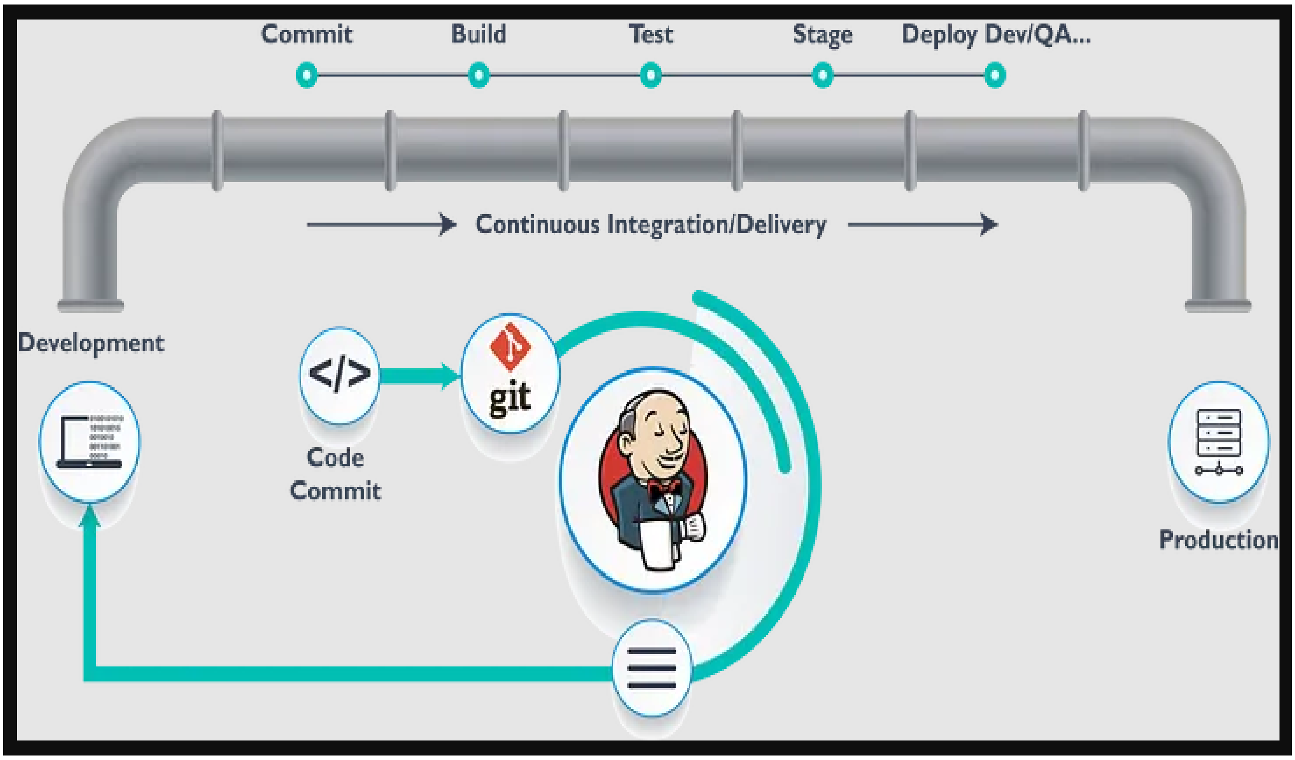
2. In larger development teams, integrating code changes from different developers can be challenging, leading to integration conflicts.

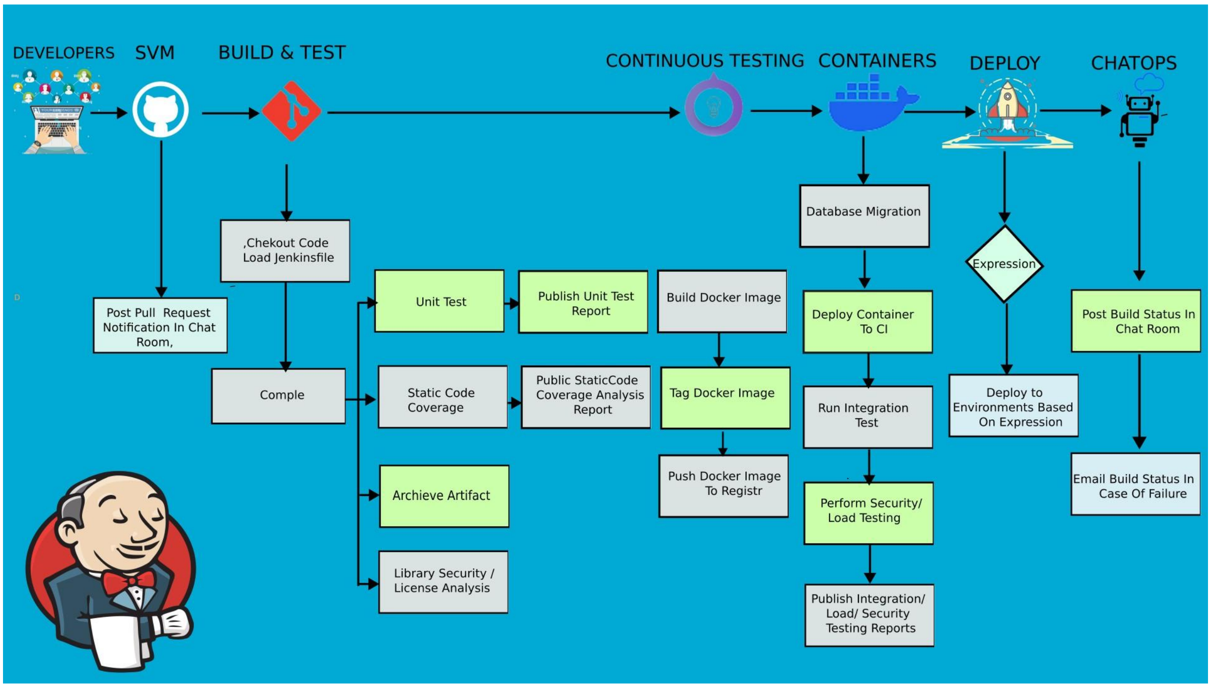
3. In case of issues with a release, rollbacks can be challenging and time-consuming.

4. Developers may face issues when deploying code to different environments due inconsistencies in configuration.

## Solutions:

CI/CD Automation using Jenkins implemented




DEVELOPERS
SVM
BUILD & TEST
CONTINUOUS TESTING
CONTAINERS
DEPLOY
CHATOPS
,Chekout Code Load Jenkinsfile
Post Pull Request Notification In Chat Room,
Comple
Unit Test
Publish Unit Test Report
Static Code Coverage
Public StaticCode Coverage Analysis Report
Archieve Artifact
Library Security / License Analysis
Build Docker Image
Tag Docker Image
Push Docker Image To Registr
Database Migration
Deploy Container To CI
Run Integration Test
Perform Security/ Load Testing
Publish Integration/ Load/ Security Testing Reports
Expression
Deploy to Environments Based On Expression
Post Build Status In Chat Room
Email Build Status In Case Of Failure

**CI/CD** is a modern DevOps practice that involves automating the build, testing, and deployment of code changes. Here are some key benefits of implementing CI/CD at the client setup:

- **Faster and More Frequent Releases:**
  - **Automation:** CI/CD pipelines automate the entire process of building, testing, and deploying code changes. This automation eliminates many manual steps and significantly reduces the time required for these processes.
  - **Confidence:** Developers can release new features and updates more frequently because they have confidence that the automated pipeline thoroughly tests and validates their code changes.
  - **Competitive Advantage:** Faster releases mean that your organization can respond quickly to market demands, stay ahead of competitors, and address customer needs more promptly.
- **Improved Code Quality:**
  - **Automated Testing:** CI/CD includes automated testing at various stages of the pipeline. This automated testing

ensures that code changes are rigorously tested for functionality, security, and performance.

- ➢ **Early Issue Detection:** By catching bugs and issues early in the development process, CI/CD helps prevent the accumulation of technical debt and reduces the cost and effort required to fix issues later.
- ➢ **Higher Confidence:** Developers and teams have higher confidence in the quality of their code, leading to more stable releases and fewer post-deployment issues.

- ❖ **Reduced Manual Errors:**
  - ➢ **Eliminating Human Error:** Automation removes the risk of human error in tasks like building, configuring, and deploying software. This leads to more reliable and consistent releases.
  - ➢ **Consistency:** With CI/CD, every code change goes through the same automated process, ensuring that releases are consistent and predictable.
- ❖ **Increased Collaboration:**
  - ➢ **Visibility:** CI/CD pipelines provide real-time visibility into the status of builds and deployments. This transparency fosters better communication and coordination among team members.
  - ➢ **Cross-Functional Teams:** CI/CD encourages collaboration between developers, testers, and operations teams, breaking down silos and promoting a more integrated approach to software development and deployment.
- ❖ **Time and Cost Savings:**
  - ➢ **Efficiency:** Automating repetitive tasks, such as building, testing, and deployment, reduces the time and effort required for these processes.
  - ➢ **Resource Optimization:** Teams can allocate their time and resources more efficiently, focusing on innovation and high-impact tasks rather than manual, time-consuming activities.

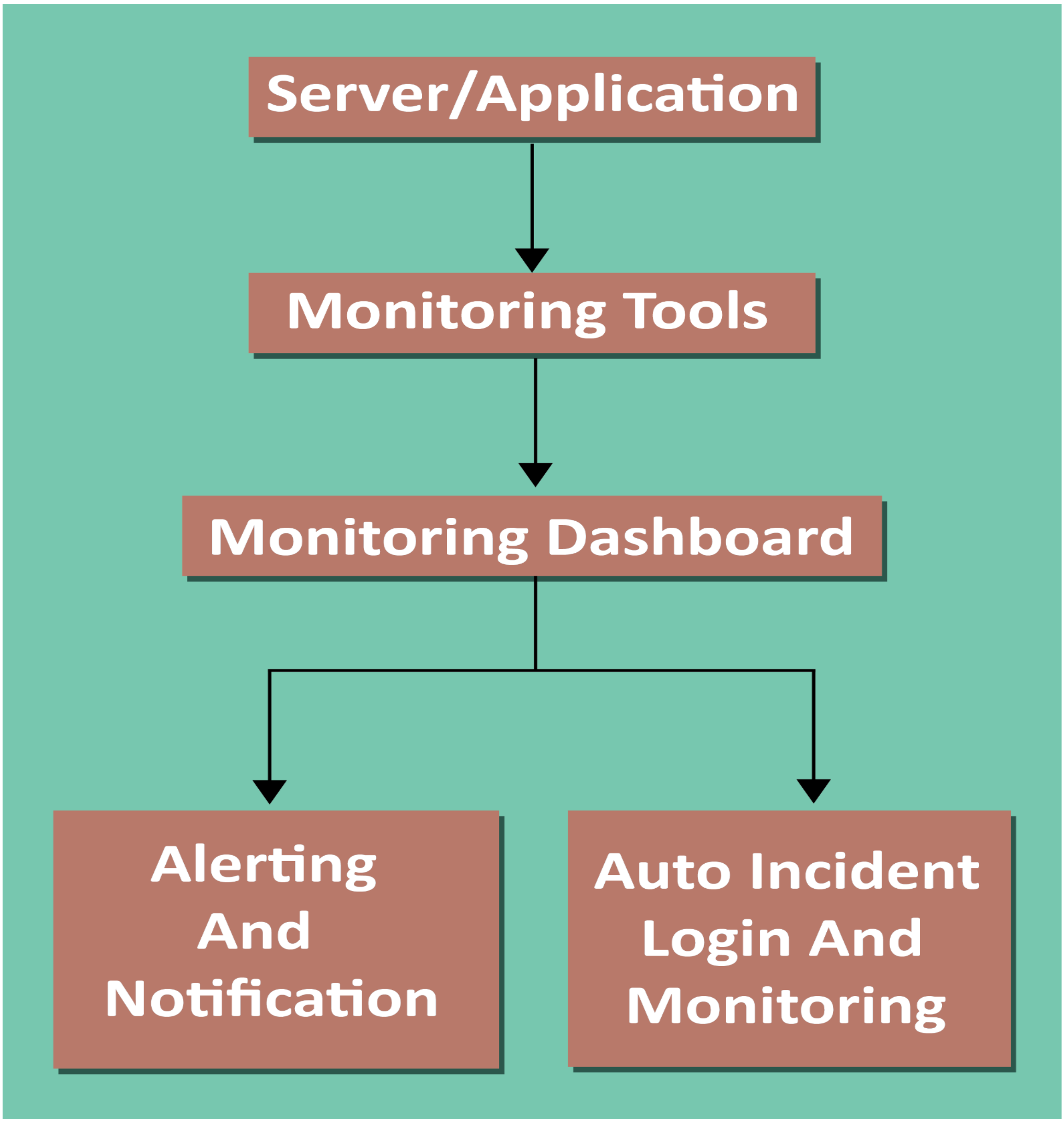
**Monitoring Setup for all servers:**

- **DB servers(connection count/load/Materialised view/backup status).**
- **Haproxy servers(load/instances status/count request)**
- **OCP servers(load/machine status)**
- **API servers(load/instance status/api performance status)**

**Problems:**

1. Monitoring required to identify the problem or service critical status and resolve it quickly.
2. Customers were not getting real-time insights into the health and performance of systems and applications, so mostly they received problems facing mail through users.
3. Problem solving was taking time because manually checking the cause of the problem will take time.
4. Users require monitoring the resources allocated across different components of the setup. This information helps in optimising resource distribution and avoiding resource conflicts.

**Solutions:**


Server/Application

Monitoring Tools

Monitoring Dashboard

Alerting And Notification

Auto Incident Login And Monitoring

**Nagios and Grafana are powerful monitoring tools** used by organizations to keep a close eye on their IT infrastructure, applications, and services. When combined, they offer a comprehensive monitoring and visualization solution. Please find below:

- **Comprehensive Monitoring**: Nagios is a robust monitoring system that allows organizations to monitor servers, network devices, services, and applications.
- **Real-time Alerting**: Nagios provides real-time alerting via email, SMS, and other channels, enabling swift response to issues and preventing downtime.
- **Centralized Dashboard**: Grafana, a data visualization tool, integrates with Nagios to create centralized dashboards, simplifying monitoring and troubleshooting.
- **Customizable Dashboards**: Grafana offers a wide range of visualization options and customization features.
- **Historical Data Analysis**: Grafana stores historical monitoring data, allowing for trend analysis and pattern identification.
- **Integration with Multiple Data Sources**: Grafana supports integration with various data sources, including databases and cloud platforms.
- **User-friendly Interface**: Grafana's intuitive interface makes monitoring data accessible to technical and non-technical users, fostering collaboration.
- **Scalability**: Nagios and Grafana are highly scalable, suitable for monitoring small to large-scale infrastructures.
- **Cost-Effectiveness**: Both Nagios and Grafana are open-source, making them cost-effective for organizations with budget constraints.

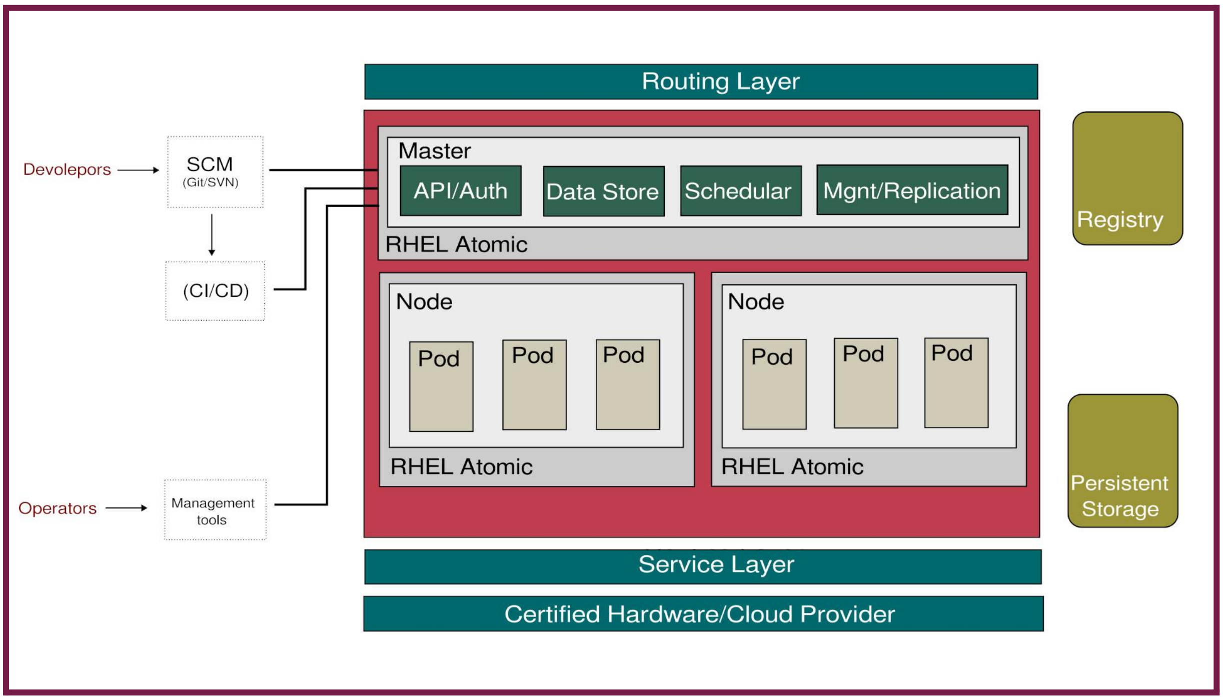
## Containerized Application Management:

- **User-Controlled Scaling and Load Balancing**
- **High Availability Requirement**
- **Desire for Autonomous Pod Scaling**

### Problems:

Users want to handle the scaling and load balancing of containers and ensure high availability of applications.
User wants pod scaling management itself as per the application load.

### Solution:



**OCP,** which stands for OpenShift Container Platform, is a Kubernetes-based enterprise container platform developed by Red Hat. It provides a comprehensive set of tools and features for deploying, managing, and scaling containerized applications. The platform offers several benefits for organizations looking to adopt containerization and cloud-native development practices. Here are some key benefits of OCP:

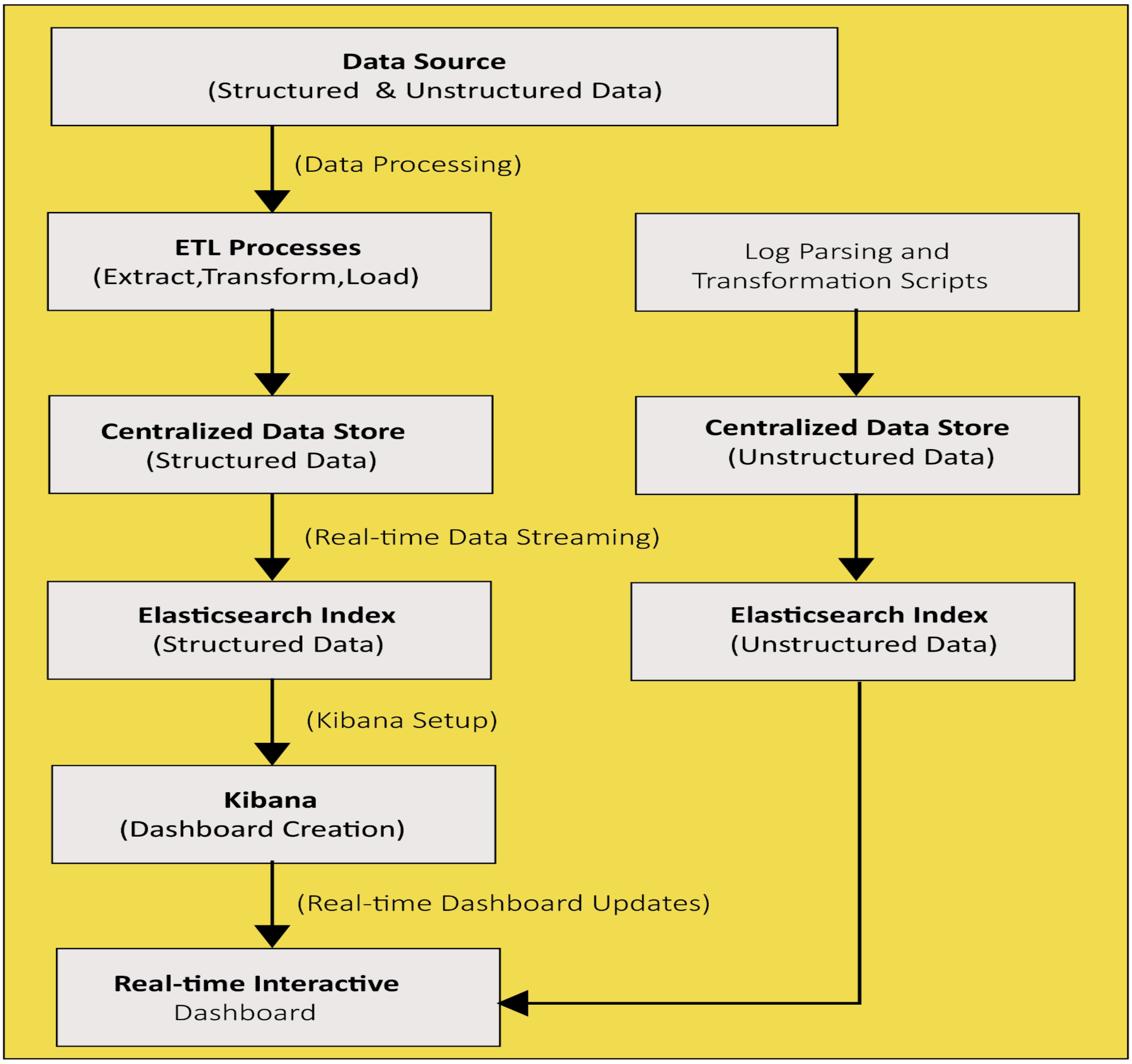
- Simplified Application Deployment: OCP abstracts the complexities of container orchestration and provides a streamlined way to deploy applications in containers. It automates the deployment process, making it easier and faster to bring applications into production.
- Scalability: OCP allows applications to scale both vertically and horizontally. It can automatically scale applications based on demand, ensuring optimal performance and resource utilization.
- Container Management: OCP provides tools to manage container lifecycles, including rolling updates, rollbacks, and automatic healing of containers. This ensures high availability and reduces downtime during updates or failures.
- DevOps Enablement: OCP fosters collaboration between development and operations teams by providing a shared platform and consistent environments for building, testing, and deploying applications. This accelerates the DevOps process.
- Application Portability: Containers deployed on OCP are highly portable. Developers can build once and deploy anywhere, whether it's on the local development environment, staging environment, or production environment.
- Monitoring and Logging: OCP offers built-in monitoring and logging capabilities to gain insights into application performance and troubleshoot issues effectively.

## Visualization and Dashboard Creation:

- **Requirement for Interactive Dashboards and Visualizations**
- **Visualization of Structured and Unstructured Data**
- **Real-time or Live Data Dashboard Representation**

**Problems:** Users require interactive dashboards and visualisations based on data in the database, which can present on kibana.
User wants to visualise structured and unstructured data as per requirement.
Users were not able to represent dashboards on the real-or-live based data.

**Solution:**



**Elasticsearch** is a powerful and widely-used open-source search and analytics engine. It is built on top of Apache Lucene and is designed to handle large-scale data processing, search, and analysis. Elasticsearch offers a wide range of benefits, making it a popular choice for various applications and use cases. Here are some key benefits of Elasticsearch:

- **Full-text Search:** Elasticsearch provides highly efficient full-text search capabilities, allowing users to perform complex text-based searches across vast amounts of data. It supports fuzzy matching, stemming, autocomplete, and various search features, making it ideal for applications that require sophisticated search functionality.
- **Real-time Data Indexing:** Elasticsearch excels at real-time indexing and analysis of data. It can ingest, index, and make data available for search and analysis in near real-time, enabling quick insights and decision-making.
- Scalability and High Availability: Elasticsearch is designed to be distributed and scalable. It can easily handle large datasets and can be clustered to ensure high availability and fault tolerance. This makes it suitable for applications with high traffic and demanding workloads.
- **JSON-based Documents:** Elasticsearch stores data in JSON format, which is widely used and easy to work with. JSON documents allow for flexible and dynamic schema mapping, making it adaptable to changing data structures.
- **Aggregation and Analytics:** Elasticsearch supports various aggregation features, allowing users to perform complex analytics and aggregations on the indexed data. It enables real-time data exploration and visualization, making it useful for business intelligence and data analytics applications.
- **Near Real-time Analytics:** With its ability to handle real-time data and powerful query capabilities, Elasticsearch is well-suited for near real-time analytics and monitoring applications. It enables users to track and analyze trends and anomalies as they happen.
- **RESTful API:** Elasticsearch provides a simple RESTful API, making it easy to interact with the system using standard HTTP methods. This API allows for easy integration with other applications and frameworks.
- **Open-source and Active Community:** Being open-source, Elasticsearch is freely available for use and has a vibrant and active community. This community continuously contributes to the platform's development, documentation, and support.
- **Multi-language Support:** Elasticsearch supports multiple programming languages through client libraries, making it accessible to developers from various language ecosystems.
- **Integration with the Elastic Stack:** Elasticsearch is a core component of the Elastic Stack, which includes Kibana (for visualization), Logstash (for data collection and processing), and Beats (for data shippers). This integration allows users to build end-to-end data pipelines for data ingestion, search, analytics, and visualization.

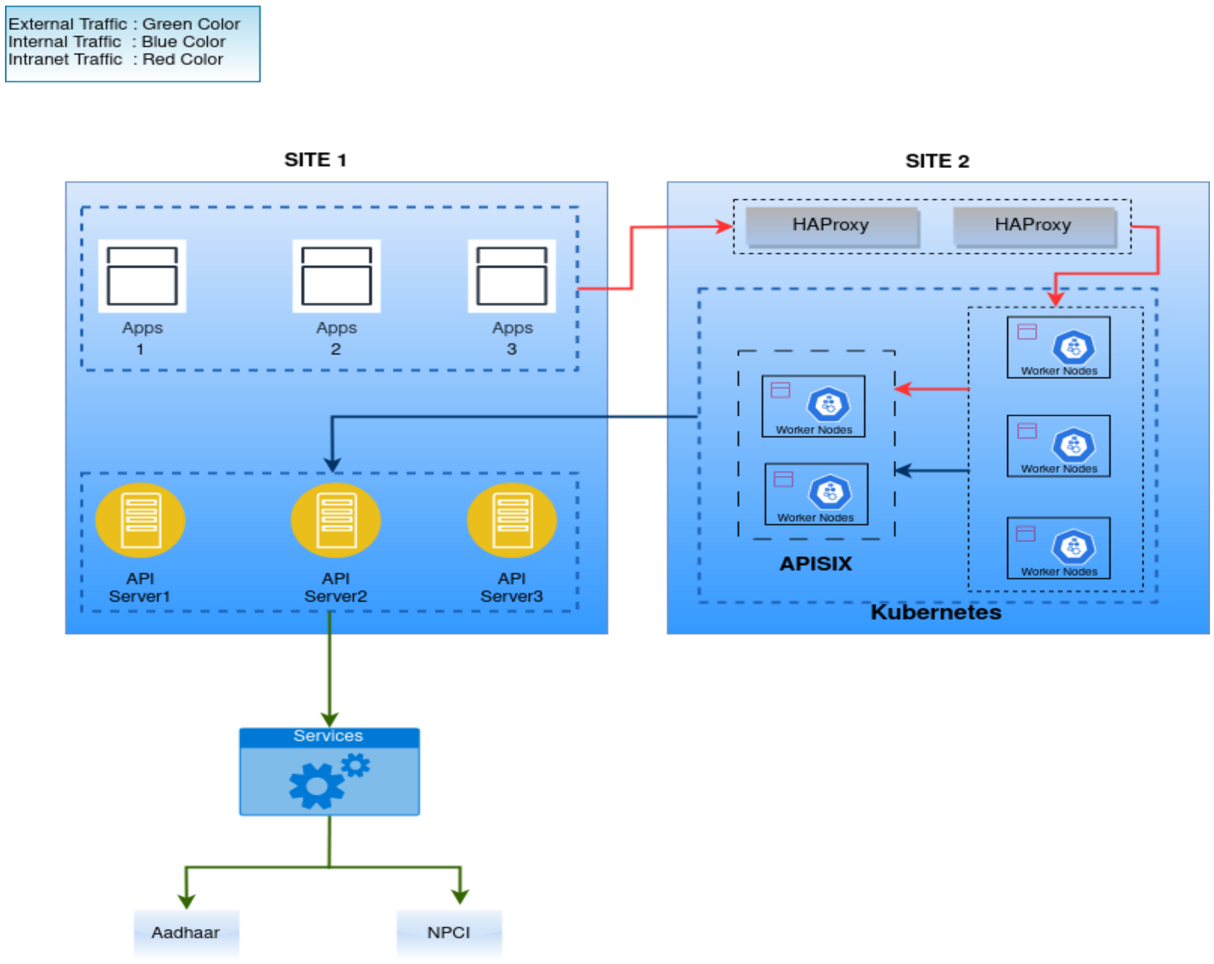
## API Management and Monitoring

- **API Gateway for Security with Authentication**
- **Request and Response Count Details**
- **Monitoring User Request Loads**

**Problems:** Customer requires API gateway for security at the time of api access with authentication.
Request and response count detail required for understanding the requirement.
User request loads also need to be monitored.

**Solution:**

Architecture of APISIX Setup



**APIsix** is a high-performance and scalable open-source API gateway that is designed to handle and manage API traffic in a distributed and efficient manner. It provides various features and functionalities that make it suitable for solving specific problems and use cases. Some of the scenarios where APIsix excels as the best tool include:

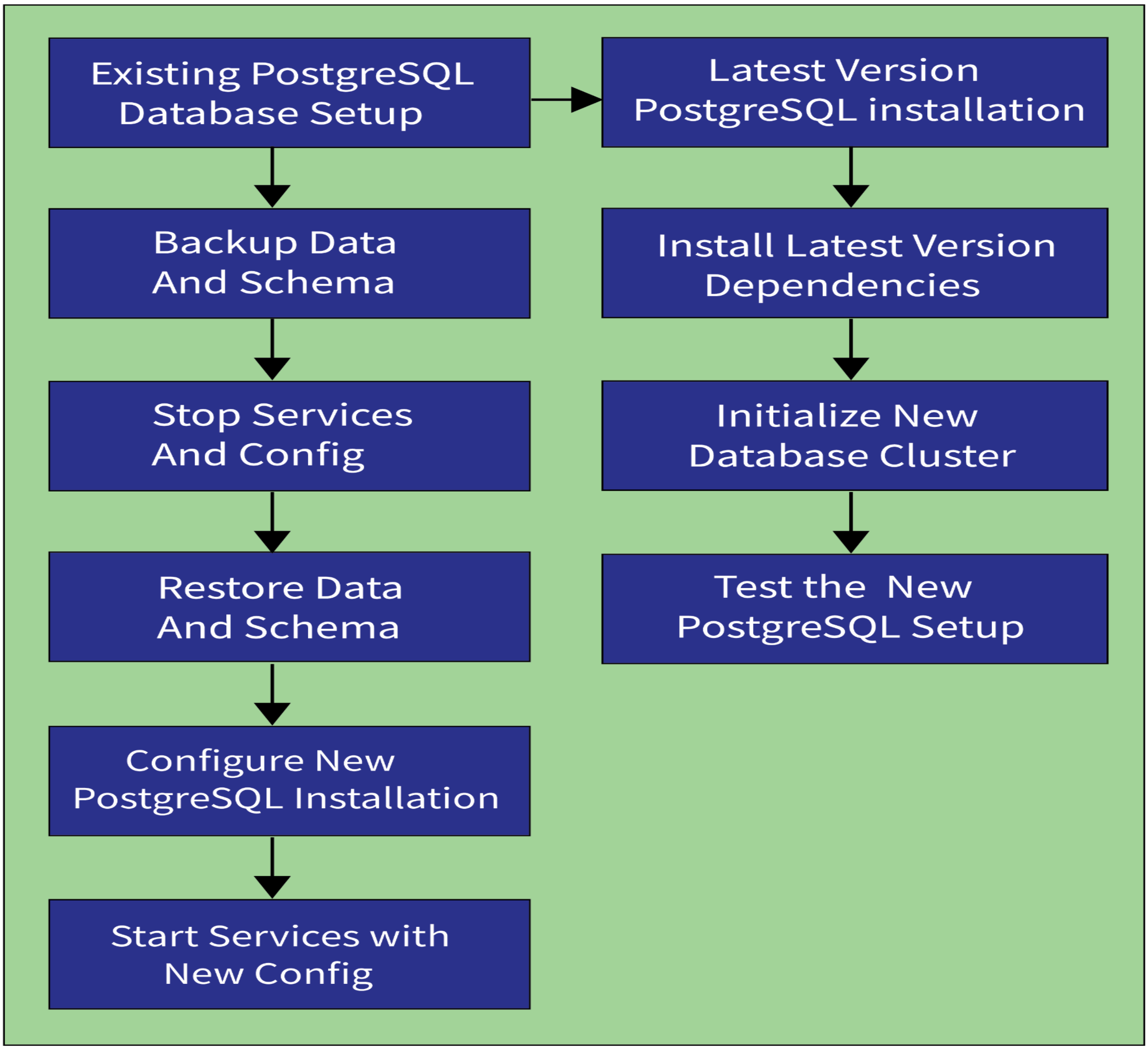
- **Microservices Architecture**: APIsix is well-suited for managing API requests and responses in a microservices-based architecture. It acts as a centralized entry point to handle incoming requests and efficiently route them to the appropriate microservices.
- **Load Balancing**: APIsix can perform load balancing across multiple backend servers or microservices instances, ensuring optimal distribution of incoming API traffic to maintain high availability and performance.
- **Security and Authentication**: APIsix provides various security features, such as API key authentication, JWT verification, and SSL termination, to ensure secure communication between clients and backend services.
- **Caching**: It supports caching responses from backend services, reducing the load on downstream systems and improving response times for frequently requested data.
- **Logging and Monitoring**: APIsix can generate detailed logs of API traffic, helping with monitoring, analytics, and debugging.
- **Analytics and Metrics**: The gateway can collect and report API usage metrics, enabling you to gain insights into API performance and usage patterns.
- **Service Discovery and Health Checks**: APIsix integrates with service discovery mechanisms and performs health checks to ensure backend services are available and healthy.
- **API Versioning**: It allows you to manage different versions of your APIs and handle version-specific requests effectively.

**PostgreSQL setup upgrade latest version:**

- **New Features**
- **Security Updates**
- **Compatibility**
- **Performance**

**Problem:** The challenge is to upgrade an existing PostgreSQL database setup to the latest available version of PostgreSQL.

**Solution:**


Existing PostgreSQL Database Setup
Backup Data And Schema
Stop Services And Config
Restore Data And Schema
Configure New PostgreSQL Installation
Start Services with New Config
Latest Version PostgreSQL installation
Install Latest Version Dependencies
Initialize New Database Cluster
Test the New PostgreSQL Setup

**PostgreSQL**, often referred to as Postgres, is a powerful open-source relational database management system (RDBMS). It is known for its advanced features, extensibility, and compliance with SQL standards. PostgreSQL supports a wide range of data types, indexing methods, and procedural languages, making it a popular choice for various applications, from small-scale projects to large-scale enterprise systems.

- **Open Source**: PostgreSQL is open-source, which means it's free to use and has a vibrant community of developers and users contributing to its development and support.
- **Advanced Features**: PostgreSQL offers advanced features like support for complex data types, full-text search, geospatial data handling, and JSON support, making it suitable for a wide range of applications.
- **Extensibility**: PostgreSQL's extensibility allows you to define custom data types, functions, and operators, making it highly adaptable to specific use cases.
- **Compliance with SQL Standards**: PostgreSQL adheres closely to SQL standards, ensuring compatibility with other SQL-based database systems and providing robust support for transactions and complex queries.
- **Scalability**: PostgreSQL can scale both vertically and horizontally, making it suitable for projects ranging from small-scale applications to large-scale enterprise systems. It can handle high workloads and provide high availability through features like replication and failover.

**HAProxy (High Availability Proxy)** is a free and open-source load balancer and proxy server software that is widely used to distribute incoming application traffic across multiple backend servers or nodes. It offers several benefits, making it a popular choice for improving the performance, reliability, and scalability of web applications. Some of the key benefits of HAProxy include:

- **Load Balancing**: HAProxy efficiently distributes incoming application traffic across multiple backend servers or nodes, ensuring optimal resource utilization and improved performance by preventing server overloads.
- **High Availability**: HAProxy continuously monitors the health of backend servers and can automatically reroute traffic away from failed or unhealthy servers, minimizing downtime and enhancing application reliability.
- **SSL/TLS Termination**: HAProxy can handle SSL/TLS encryption and decryption, relieving backend servers of this resource-intensive task and improving overall performance, especially for secure web applications.
- **Reverse Proxy Capabilities**: HAProxy acts as a powerful reverse proxy, offering advanced routing and optimization features such as request and

response manipulation, header management, and content caching, enhancing the flexibility and performance of web applications.

- **Granular Traffic Control**: HAProxy provides fine-grained control over traffic routing, allowing administrators to define rules based on factors like URL paths, headers, cookies, and source IP addresses. This enables precise management of application traffic.