A project report on

APPLICATION AND DATABASE ORCHESTRATION USING KUBERNETES (TEAM 35)

From

Drive ready plus(DevOps)

Submitted by

20P31A0537
20P31A0538
20P31A0546
20P31A0589
20P31A0592

Under the supervision of

BOBBY PAMARTHI

SURYA ASHOK

TECHNICAL HUB

ADITYA ENGINEERING COLLEGES

SURAMPALEM,ANDHRA PRADESH.

SCOPE OF THE PROJECT:

The scope of application and database orchestration using Kubernetes can vary depending on the specific requirements of the application and database. Some of the common use cases for Kubernetes in database orchestration include:

1. **Deploying databases as part of a microservices architecture:** Microservices are small, independent services that communicate with each other to form an application. Deploying databases as part of a microservices architecture can help improve scalability, resilience, and performance.
2. **Running multiple database instances:** Kubernetes can be used to deploy multiple database instances, each with its own set of resources and configuration. This can help improve availability and scalability.
3. **Automating database backups and restores:** Kubernetes can be used to automate the process of backing up and restoring databases. This can help reduce the risk of data loss and downtime.
4. **Scaling databases based on demand:** Kubernetes can be used to scale databases up or down based on demand. This can help ensure that the database can handle increased traffic and workload.
5. **Deploying databases across multiple clusters:** Kubernetes can be used to deploy databases across multiple clusters, either on-premises or in the cloud. This can help improve availability and reduce latency.

Overall, the scope of application and database orchestration using Kubernetes is broad and can include a range of use cases depending on the specific requirements of the application and database.

PURPOSE OF THE PROJECT:

The purpose of application and database orchestration using Kubernetes is to simplify the deployment, scaling, and management of applications and databases running in containers. Kubernetes provides a framework for automating many of the tasks involved in managing large, distributed container deployments.

Some of the key purposes of application and database orchestration using Kubernetes include:

1. **Simplified deployment:** Kubernetes allows for easy deployment of containerized applications and databases across a distributed environment. It provides a consistent platform for deploying applications and databases, regardless of the underlying infrastructure.
2. **Automatic scaling:** Kubernetes can automatically scale applications and databases based on resource usage and demand. This helps to optimise resource utilisation and ensure high availability.
3. **Load balancing:** Kubernetes provides built-in load balancing to distribute traffic across multiple instances of an application or database, ensuring that requests are processed efficiently.

4. **Self-healing:** Kubernetes is designed to detect and respond to failures automatically. If an application or database instance fails, Kubernetes can automatically replace it with a new instance.
5. **Portability:** Kubernetes is designed to be cloud-agnostic, meaning that it can run on any cloud provider or on-premises infrastructure. This makes it easier to deploy and manage containerized applications and databases across different environments.

TOOLS USED:

There are various tools and technologies used for application and database orchestration using Kubernetes. Some of the commonly used tools are:

Kubernetes Dashboard: Kubernetes dashboard is a web-based user interface that allows users to deploy, scale, and manage containerized applications and databases running in Kubernetes clusters.

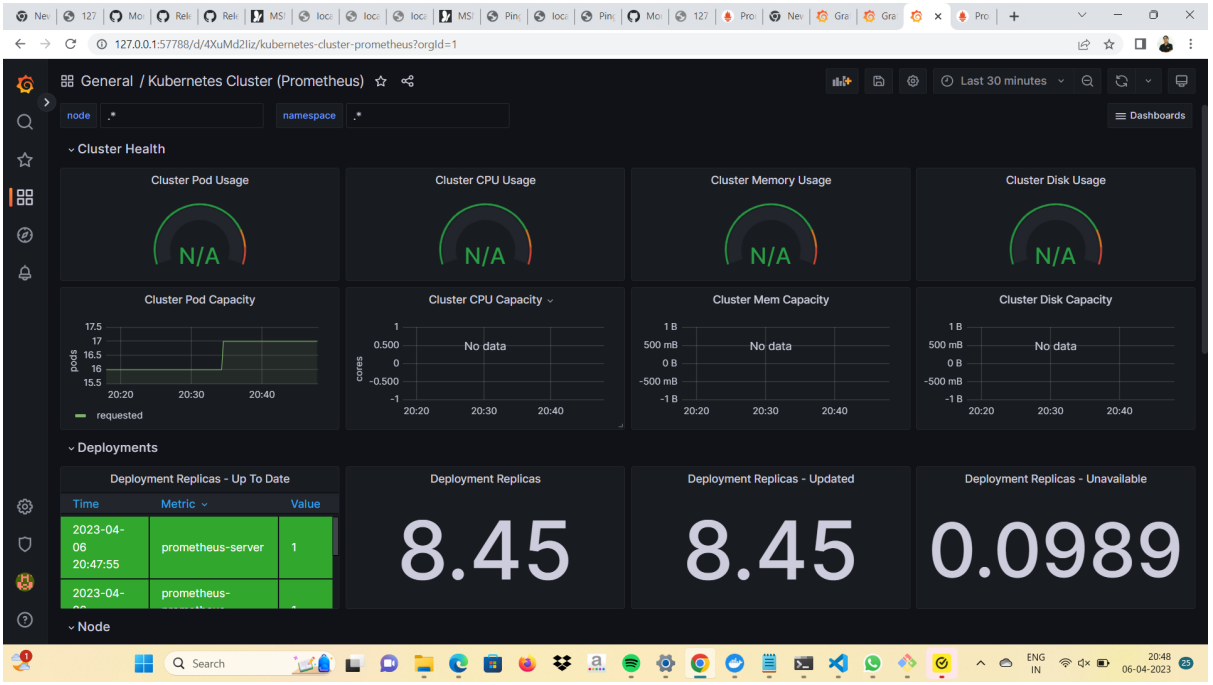
Kubectl: Kubectl is a command-line tool that allows users to interact with Kubernetes clusters. It is used to deploy, manage, and troubleshoot applications and databases running in Kubernetes clusters.

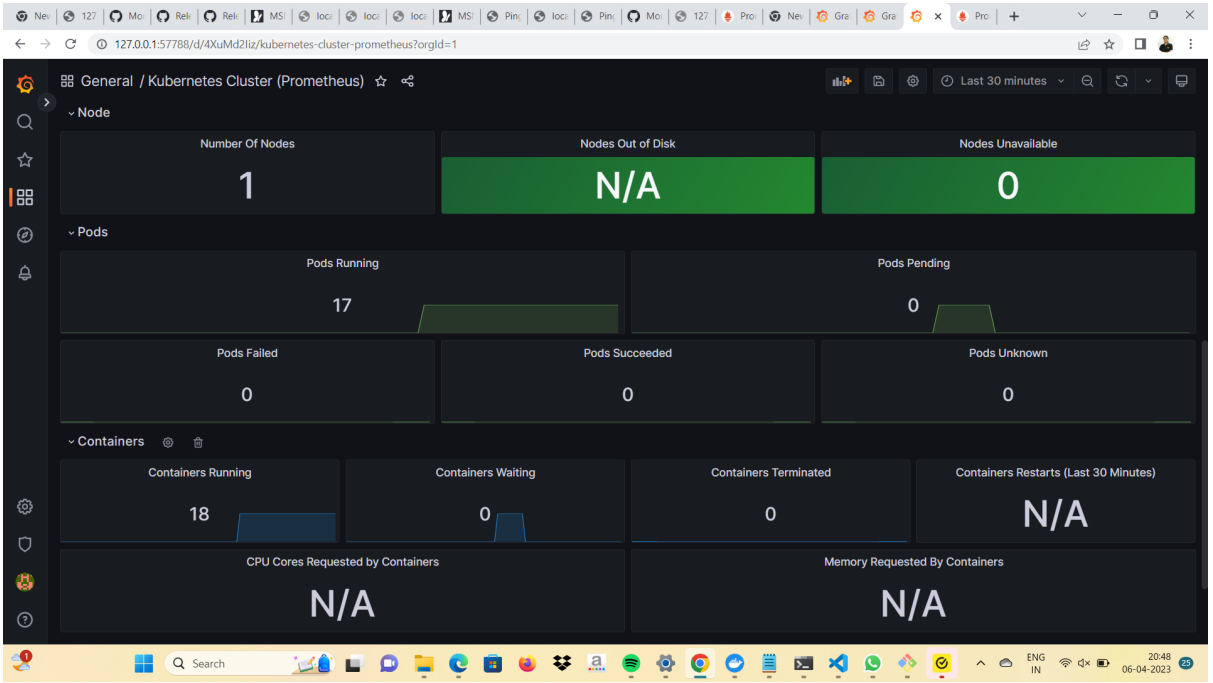
Helm: Helm is a package manager for Kubernetes that simplifies the deployment and management of containerized applications and databases. It allows users to package, deploy, and manage Kubernetes applications using predefined charts.

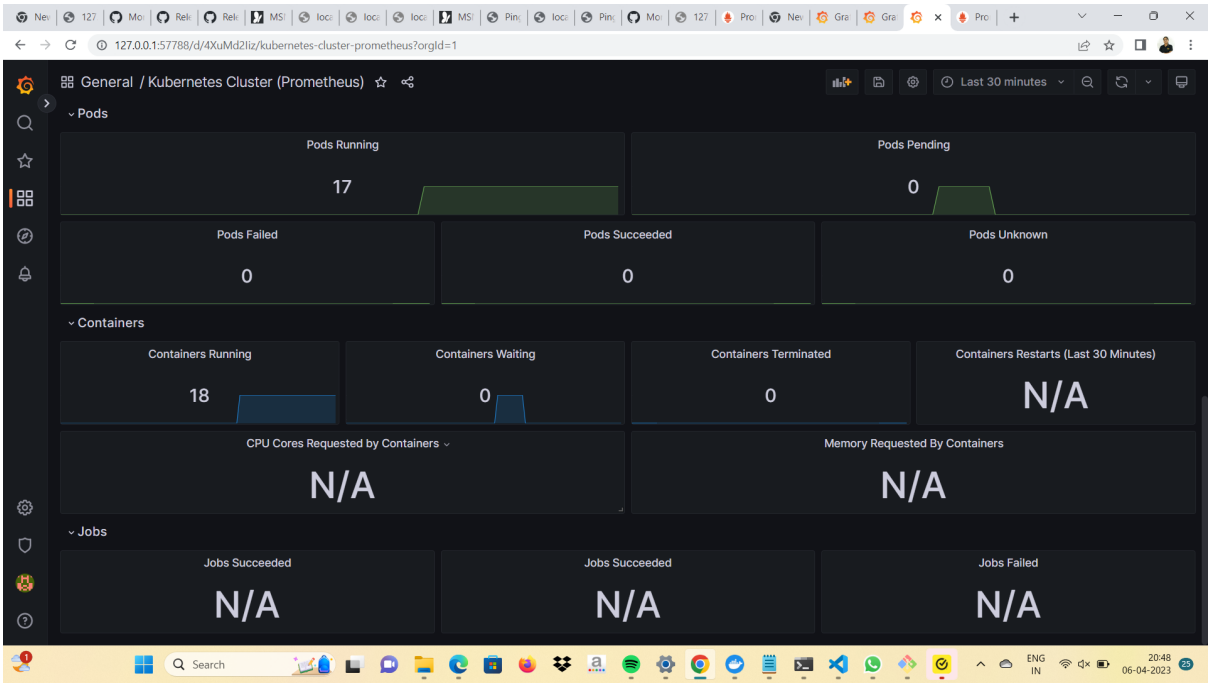
Prometheus: Prometheus is a monitoring and alerting toolkit for Kubernetes that provides a framework for monitoring containerized applications and databases running in Kubernetes clusters.

Grafana: Grafana is an open-source dashboard and visualisation platform that allows users to monitor and visualise data from various sources, including Kubernetes clusters and Prometheus.

PROOF OF CONCEPT:







REPOSITORY DETAILS:

[KUBERNETES COMMANDS]

OUTCOME:

Using orchestration with Kubernetes to deploy and manage an application and database can have several outcomes, including:

Improved Scalability: Kubernetes enables you to scale your application and database resources up or down dynamically based on the workload, making it easier to handle traffic spikes or increased demand without affecting performance.

High Availability: Kubernetes ensures that your application and database remain available even if one or more nodes fail. This is

achieved by automatically detecting failures and replacing or rescheduling the affected pods to run on healthy nodes.

Efficient Resource Utilisation: Kubernetes optimises resource utilisation by managing containers and pods efficiently, ensuring that resources are not wasted.

Simplified Deployment and Management: Kubernetes automates the deployment, scaling, and management of your application and database, making it easier to manage and reducing the risk of errors.

Increased Flexibility: Kubernetes provides a high level of flexibility and customization, enabling you to choose the best configuration for your application and database.

Overall, using orchestration with Kubernetes can lead to improved efficiency, scalability, and availability of your application and database, making it a popular choice for modern cloud-based architectures.

CONCLUSION:

In conclusion, Kubernetes is an excellent tool for orchestrating applications and databases. It provides a wide range of features and capabilities for managing containerized applications and databases in a distributed environment.

By using Kubernetes, developers and system administrators can easily automate many aspects of application and database deployment, scaling, and management, which can save time, reduce errors, and improve overall efficiency.

Kubernetes provides advanced features such as auto-scaling, load balancing, rolling updates, and self-healing capabilities, which can help ensure that applications and databases are always available, scalable,

and performant. With Kubernetes, it's also easy to add or remove nodes to the cluster to accommodate changing workload demands.

In addition, Kubernetes provides a rich ecosystem of tools and plugins that can be used to customise and extend the platform to meet specific needs. This makes it easy to integrate with other systems and services, such as monitoring, logging, and security tools.

Overall, Kubernetes provides a powerful and flexible platform for orchestrating applications and databases in a distributed environment. By using Kubernetes, organisations can improve their overall agility, scalability, and reliability while reducing the complexity and cost of managing their applications and databases.