

10 Kubernetes Hands On Labs to Ace Your Interview



Why Master Kubernetes?

Mastering Kubernetes helps you manage and scale applications more easily. It automates many tasks, making your work faster and more efficient.

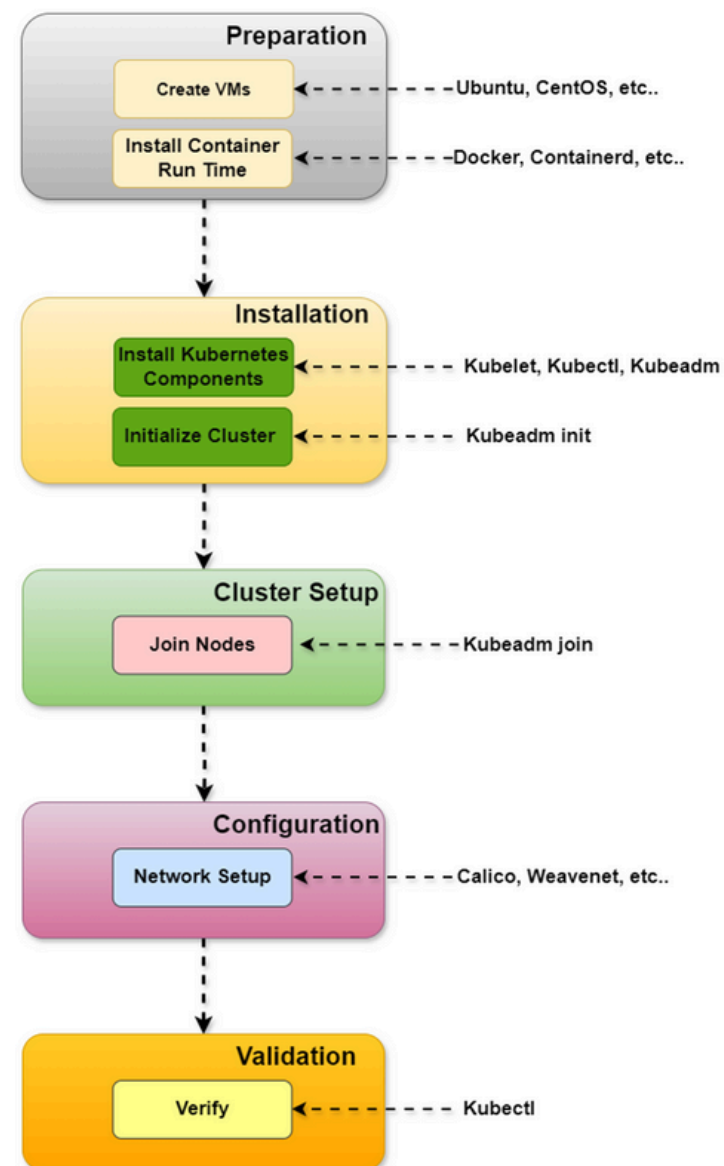
Knowing Kubernetes is valuable for jobs in tech, especially in cloud and DevOps, and can boost your career by showing you can handle complex systems

Lab 1: Setting up a Kubernetes Cluster

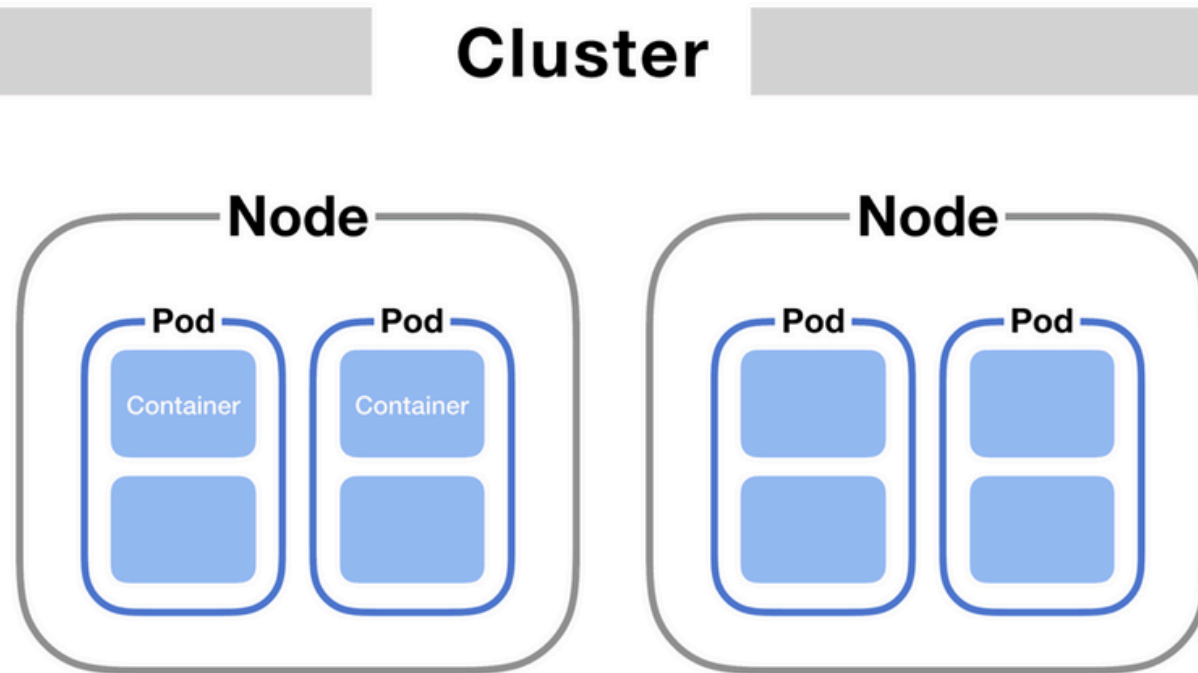
Objective: Understand the fundamentals of setting up a Kubernetes cluster

Steps:

- Provision virtual machines or cloud instances.
- Install Kubernetes components (kubectl, kubeadm, etc.).
- Initialize the master node and join worker nodes.



Lab 2: Deploying Applications on Kubernetes

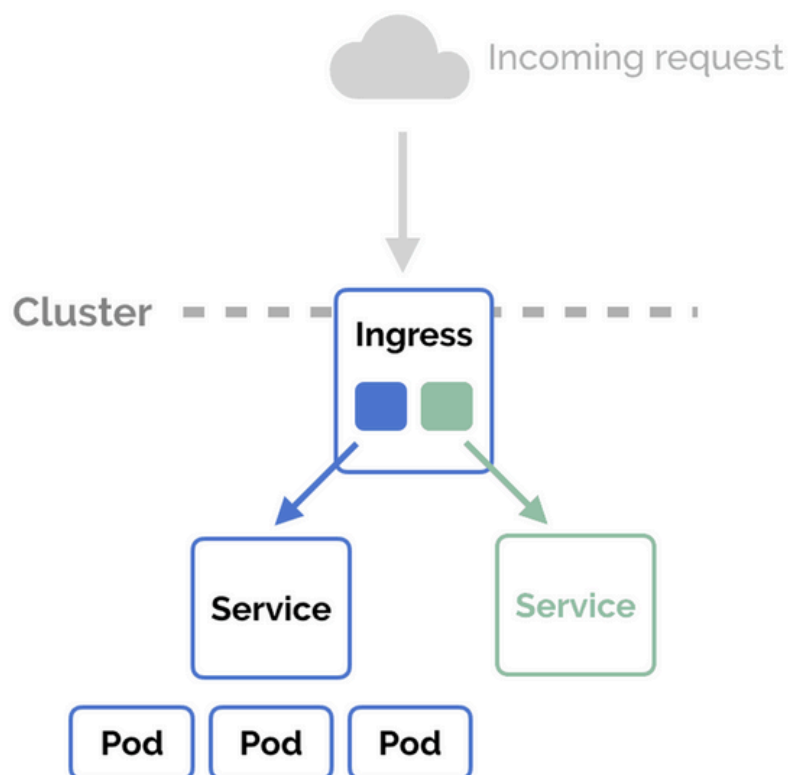


Objective: Learn how to deploy containerized applications on Kubernetes.

Steps:

- Create a Docker image of your application.
- Define deployment manifests in YAML.
- Use `kubectl apply` to deploy the application.

Lab 3: Working with Services and Ingress

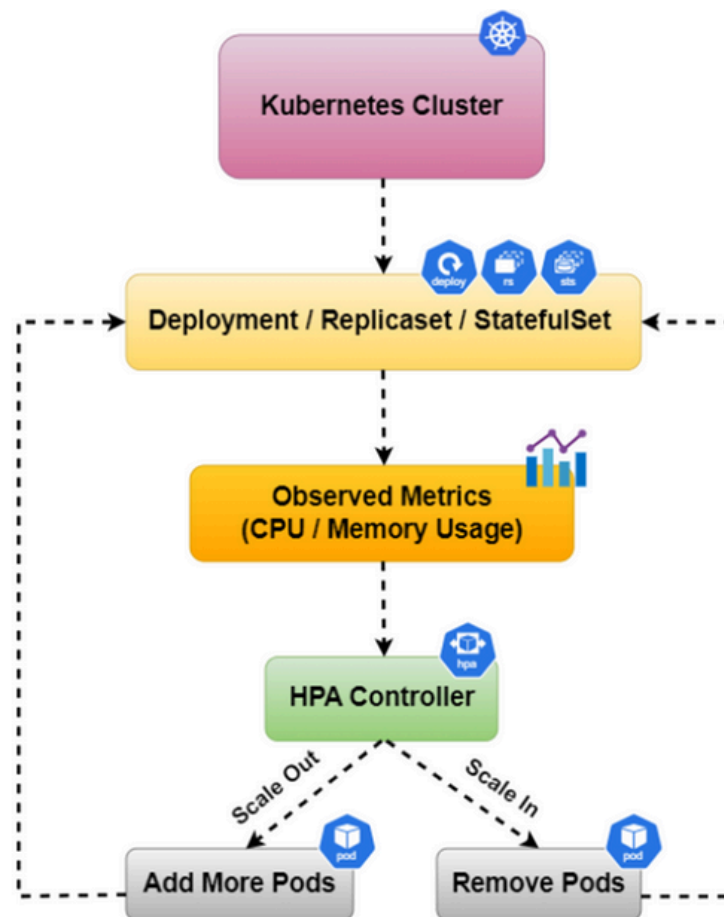


Objective: Learn how to expose applications within and outside the cluster.

Steps:

- Define a Kubernetes Service to expose your application internally.
- Create an Ingress resource to manage external access.
- Configure Ingress controllers for load balancing.

Lab 4: Scaling Applications

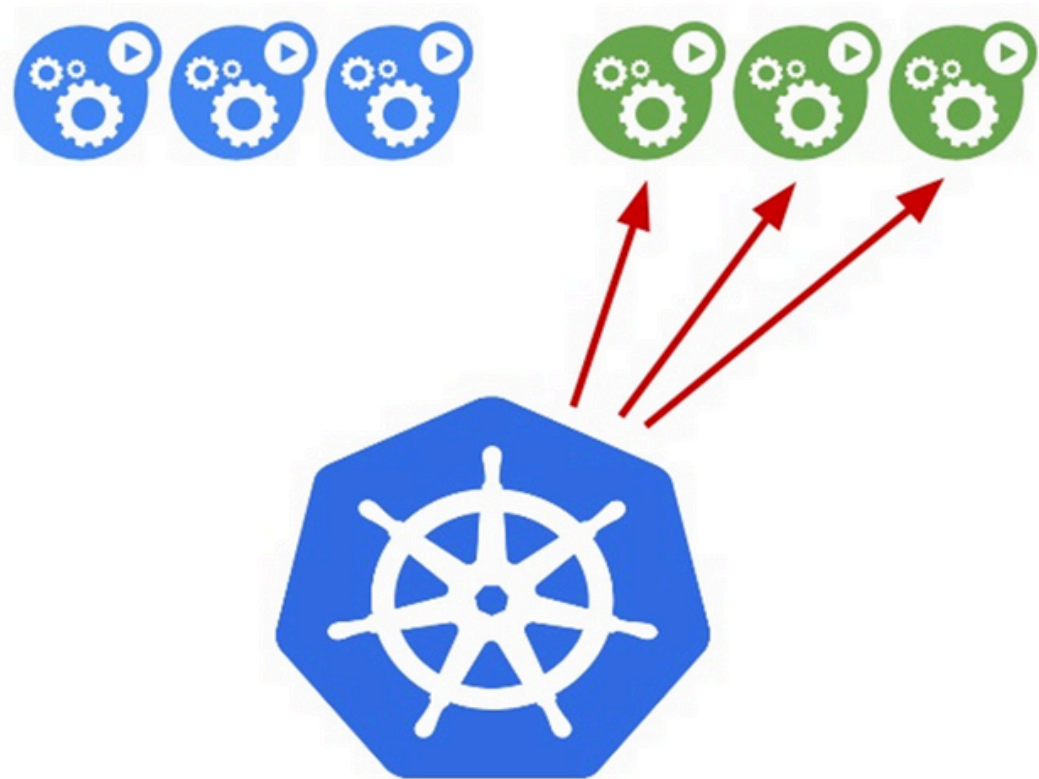


Objective: Learn to scale applications horizontally and vertically.

Steps:

- Use the `kubectl scale` to increase or decrease the number of Pods.
- Implement auto-scaling based on CPU/memory usage.
- Scale up resources like CPU and memory for individual Pods.

Lab 5: Rolling Updates and Rollbacks

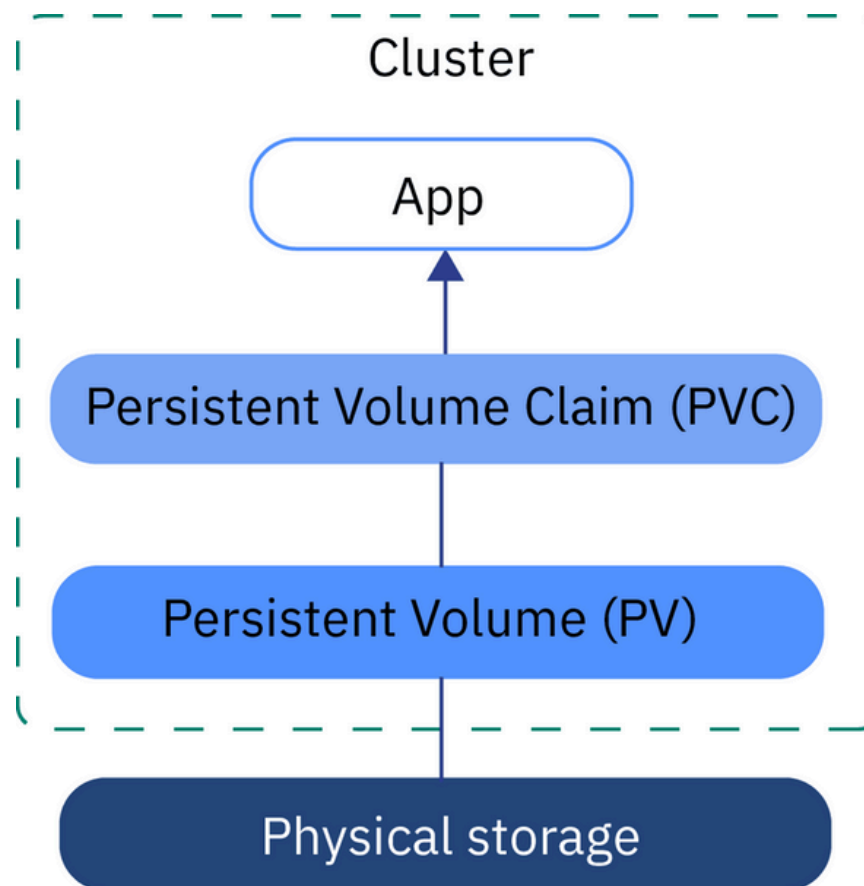


Objective: Manage application updates with minimal downtime

Steps:

- Deploy a new version of your application using a rolling update.
- Monitor the update process for errors or issues.
- Roll back to a previous version if the update fails.

Lab 6: Persistent Storage in Kubernetes

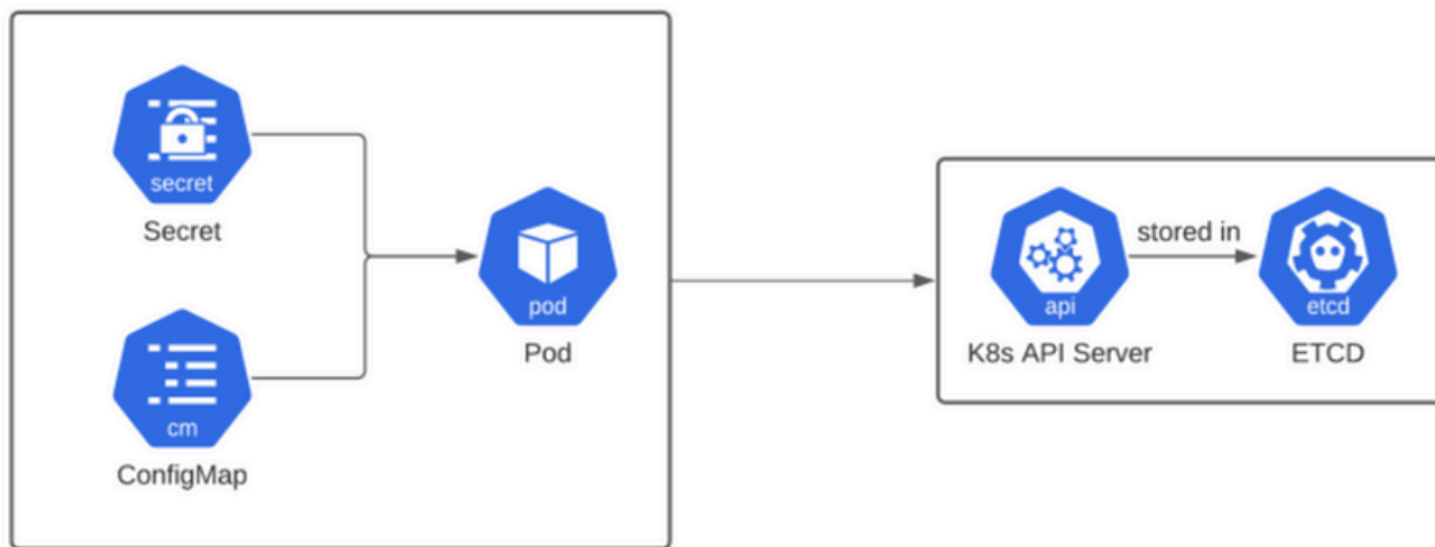


Objective: Learn how to use persistent storage for stateful applications.

Steps:

- Create a PersistentVolume (PV) and PersistentVolumeClaim (PVC).
- Attach PVCs to Pods to persist data.
- Explore different storage classes for dynamic provisioning.

Lab 7: ConfigMaps and Secrets

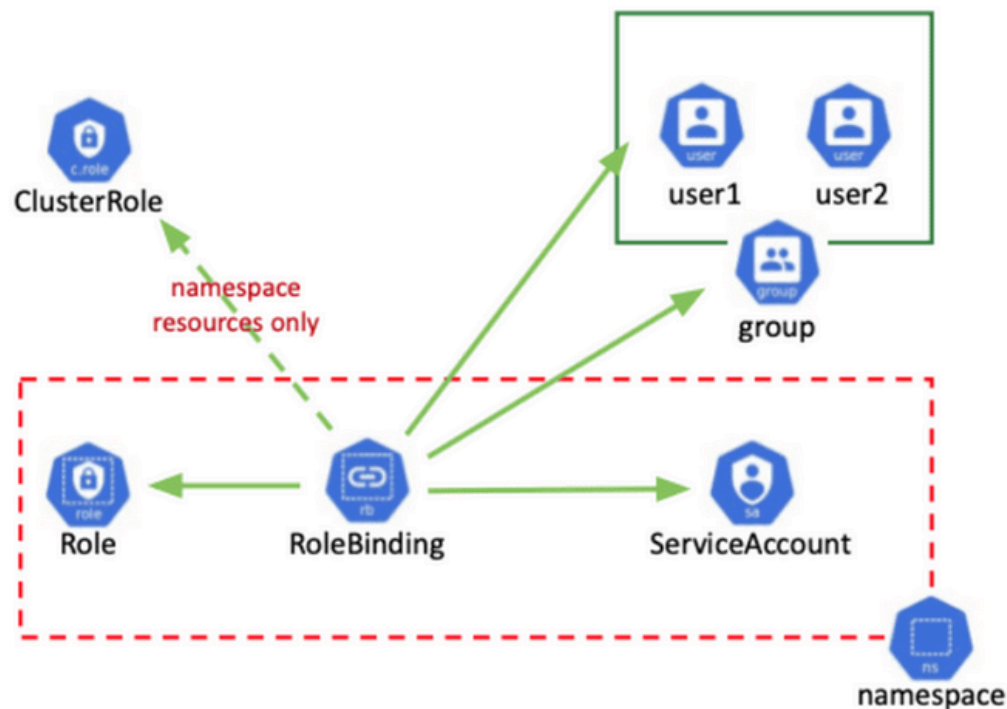


Objective: Understand how to manage configuration data and sensitive information in Kubernetes.

Steps:

- Create a ConfigMap and mount it as a volume in a Pod.
- Create a Secret and use it to store sensitive data like passwords.
- Inject ConfigMaps and Secrets into Pods using environment variables.

Lab 8: Securing Your Kubernetes Cluster

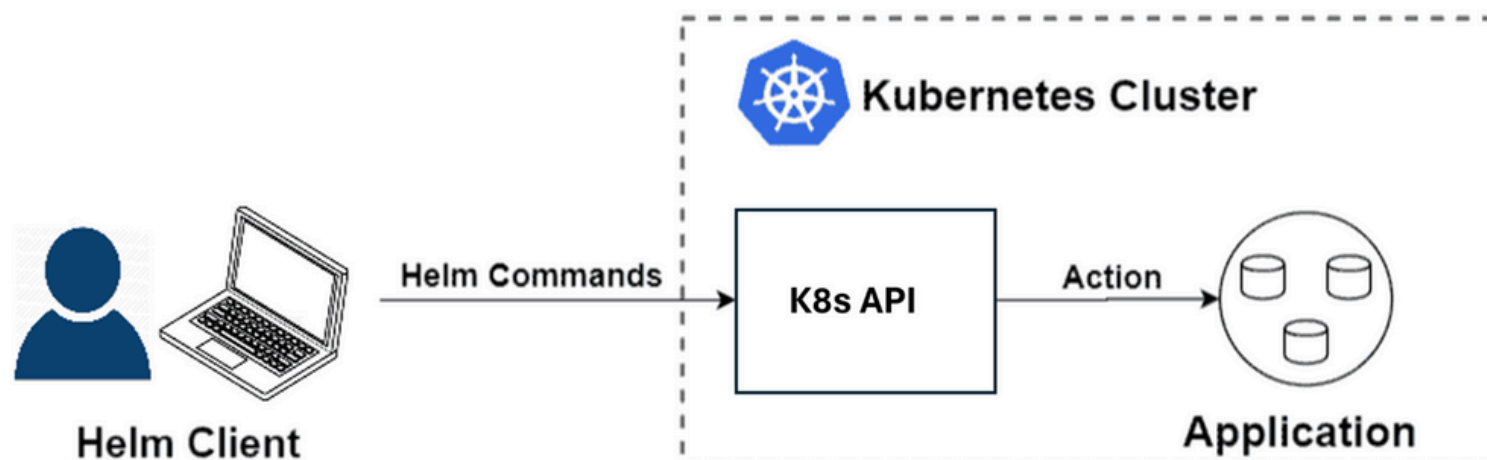


Objective: Apply security best practices to protect your Kubernetes environment

Steps:

- Implement role-based access control (RBAC) for user permissions.
- Secure API communication using TLS certificates.
- Apply network policies to control traffic between Pods.

Lab 9: Using Helm for Package Management

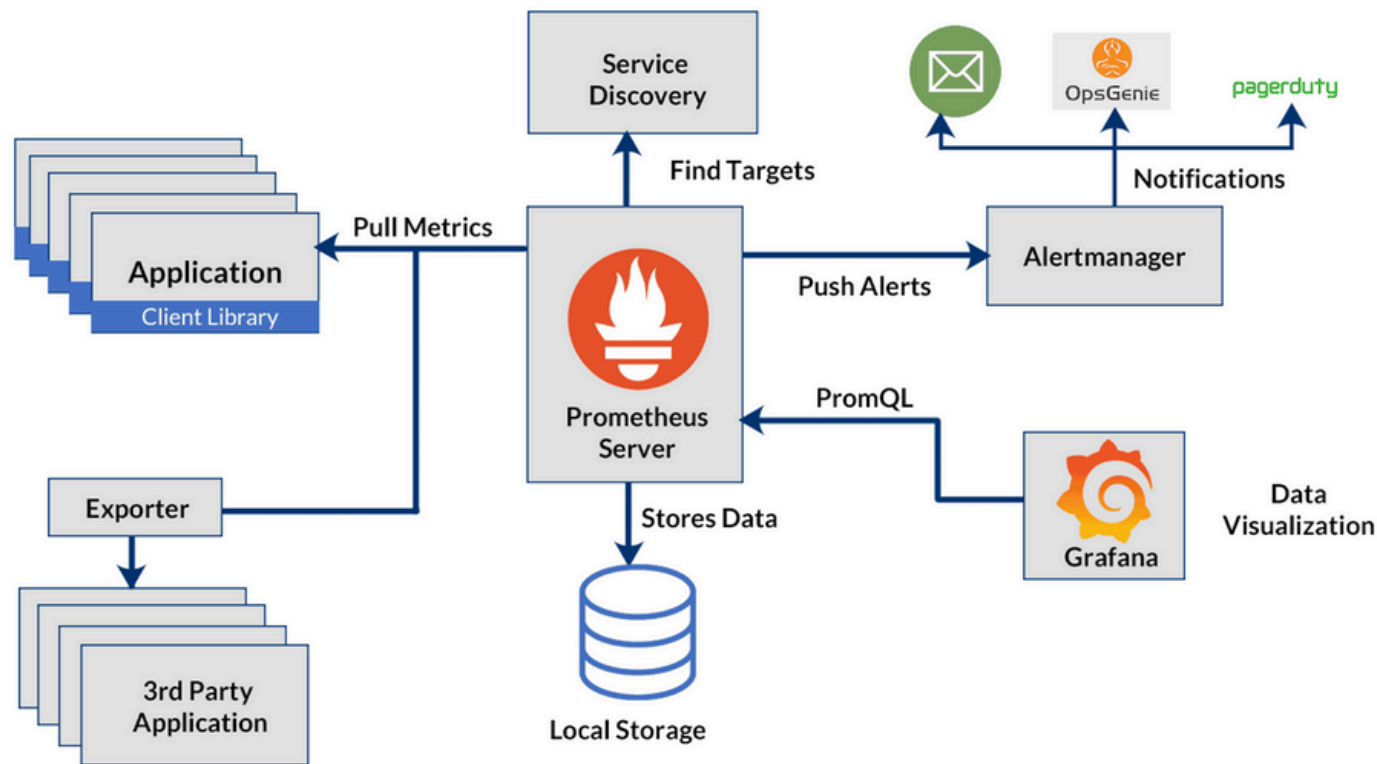


Objective: Simplify the deployment of Kubernetes applications using Helm

Steps:

- Install Helm and configure the Helm client.
- Deploy an application using a Helm chart.
- Customize the deployment by modifying Helm values.

Lab 10: Monitoring and Logging



Objective: Implement monitoring and logging to keep track of cluster health.

Steps:

- Deploy a monitoring stack using Prometheus and Grafana.
- Set up logging with Fluentd and Elasticsearch.
- Create alerts based on resource usage or application errors.