KUBERNETES
NOTES

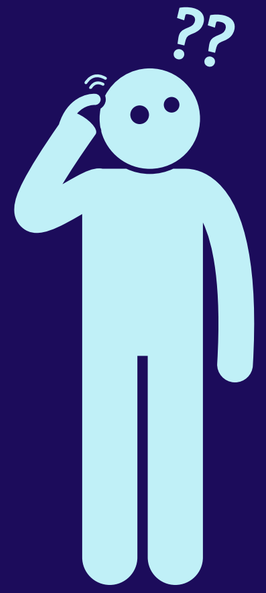
Assma Fadhli
@SpectrumShift DevOps Odyssey

WHAT DOES "K8S" MEAN?

K8S IS A NUMERONYM FOR KUBERNETES:

- K = first letter
- s = last letter
- 8 = number of letters in between (u, b, e, r, n, e, t, e)

⇢ So: K + 8 letters + s = K8s



Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

WHAT IS KUBERNETES?

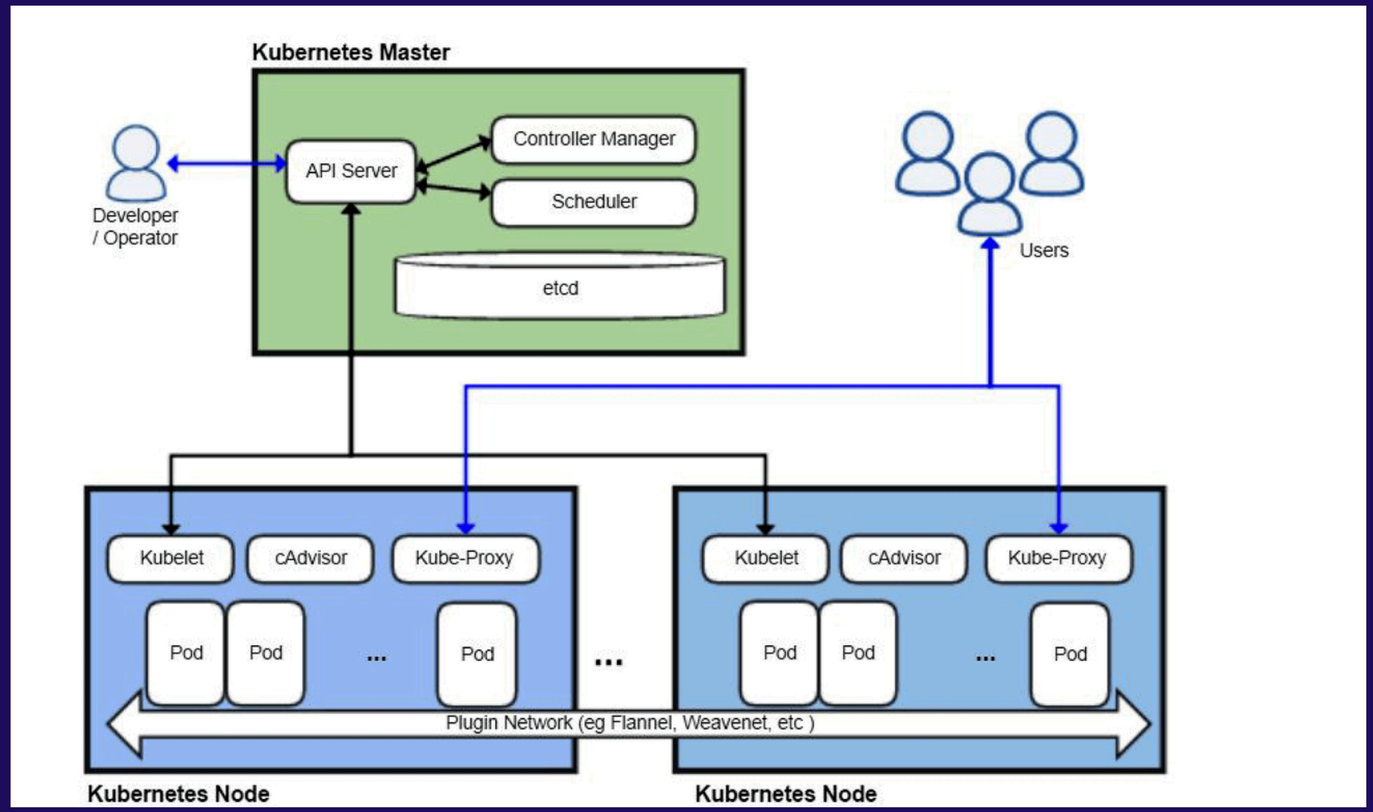
Kubernetes (K8s) is an open-source platform for automating the deployment, scaling, and management of containerized applications.

- Runs distributed systems reliably
- Uses declarative configuration
- Widely used for container orchestration

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBERNETES ARCHITECTURE OVERVIEW



Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBERNETES ARCHITECTURE OVERVIEW

Kubernetes Architecture defines how the platform manages containerized applications across a cluster of machines using a control plane and worker nodes.

- Master: Manages the cluster via the API Server, Scheduler, Controller Manager, and etcd for storing state
- Node: Runs workloads using Kubelet, Kube-Proxy, cAdvisor, and Pods (containers)
- Networking: Plugins like Flannel enable communication between pods across nodes
- Access: Developers interact via the API; users reach apps through exposed services

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBERNETES CONTROL PLANE

The Control Plane is the brain of Kubernetes.

- Coordinates cluster activities such as scheduling, scaling, and updates.
- Key components include API Server, Controller Manager, Scheduler, and etcd.
- Communicates with worker nodes to maintain desired cluster state.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

API SERVER

The API Server is the front end of the Kubernetes Control Plane.

- Acts as the gateway for all REST commands used to control the cluster.
- Processes and validates API requests.
- Provides a unified interface to connect clients, users, and components.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CONTROLLER MANAGER

The Controller Manager runs controllers that regulate the cluster state.

- Each controller is a loop that monitors the state of your cluster.
- Ensures that the current state of the system matches the desired state.
- Examples include node controllers, replication controllers, and endpoints controllers.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

SCHEDULER

The Kubernetes Scheduler assigns work to nodes.

- Watches for newly created Pods with no assigned node and selects a suitable node for them.
- Considers factors such as resource requirements, quality of service, and hardware constraints.
- Plays a crucial role in ensuring optimal resource utilization.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

ETCD

etcd is a distributed key-value store that Kubernetes uses for all cluster data.

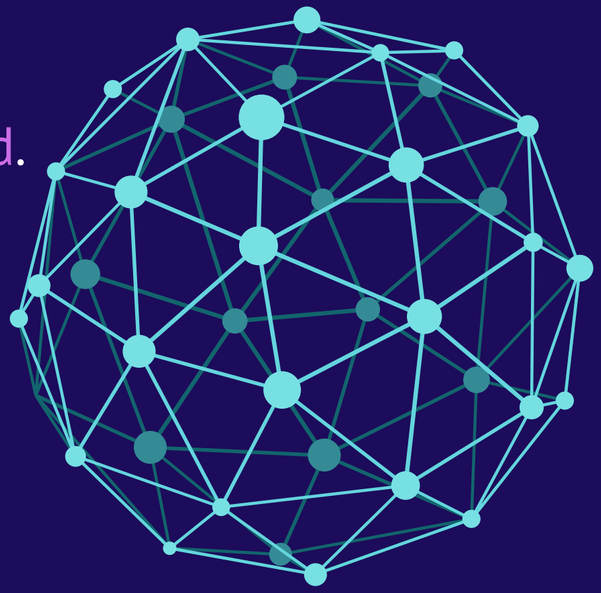
- Serves as the cluster's source of truth.
- Stores configuration data, state, and metadata reliably.
- Uses a consistent and highly available design, ensuring data integrity.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBERNETES NODES (WORKER NODES)

Worker Nodes are the machines where containers are deployed.



- Run the necessary services to support running Pods.
- Communicate with the Control Plane via the kubelet.
- Can be physical or virtual machines.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

THE KUBELET

The kubelet is an agent running on each worker node.

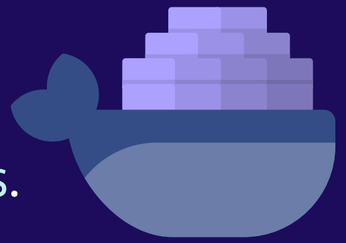
- Responsible for ensuring that containers are running in a Pod.
- Communicates with the API Server to report node and pod status.
- Manages container lifecycle using specifications defined in Pod manifests.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CONTAINER RUNTIME

Kubernetes supports multiple container runtimes (Docker, containerd, CRI-O).



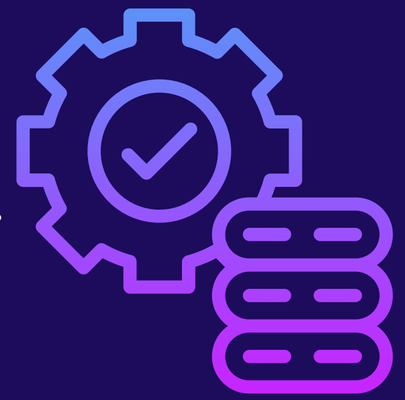
- Responsible for running containers.
- Interfaces with the kubelet to execute and manage container workloads.
- Ensures that container images are run as specified by the user.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBE-PROXY

Kube-Proxy handles networking within the cluster.



- Maintains network rules on nodes to enable communication to Pods.
- Supports different proxy modes (iptables, IPVS).
- Balances network traffic across different pods.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

PODS

A Pod is the smallest deployable unit in Kubernetes.

- Consists of one or more containers that share storage, network, and a specification for how to run the containers.
- Represents a single instance of a running process.
- Often used as building blocks for higher-level abstractions.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

SERVICES IN KUBERNETES

Services provide stable endpoints to access Pods.

- Abstract away the ephemeral nature of individual pods.
- Allow communication between internal and external components of the application.
- Types include ClusterIP, NodePort, LoadBalancer, and ExternalName.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

REPLICASETS

ReplicaSets ensure that a specified number of pod replicas are running at any given time.

- Automatically adds or removes pods to achieve the desired state.
- Provides high availability for application instances.
- Often managed indirectly by Deployments.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

DEPLOYMENTS

Deployments provide declarative updates for Pods and ReplicaSets.

- Define the desired state of your application.
- Allow rolling updates, rollbacks, and scaling.
- Simplify management of application life cycles.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

STATEFULSETS

StatefulSets manage stateful applications with unique, persistent identities.

- Guarantees the ordering and uniqueness of pods.
- Ideal for distributed systems like databases that require stable network identifiers.
- Provides persistent storage management for each pod.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

DAEMONSETS

DaemonSets ensure that a copy of a pod runs on all (or some) nodes in the cluster.

- Useful for deploying background tasks or monitoring agents.
- Automatically adjusts to cluster changes (e.g., new node arrivals).
- Ensures consistency across the cluster.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

JOBS

Jobs are used to run short-lived, non-continuous tasks to completion.

- Ensure that a specified number of pods terminate successfully.
- Suitable for batch processing and parallelized work.
- Can handle failures by restarting pods until completion criteria are met.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CRONJOBS

CronJobs schedule Jobs to run at specified times or intervals.

- Similar to cron in Unix-like systems.
- Useful for periodic and recurring tasks like backups or reports.
- Automatically manages job creation based on a defined schedule.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

INGRESS

Ingress manages external access to the services in a Kubernetes cluster.

- Provides HTTP and HTTPS routing to services.
- Can offer load balancing, SSL termination, and name-based virtual hosting.
- Operates through Ingress controllers that handle requests.

Read More >>

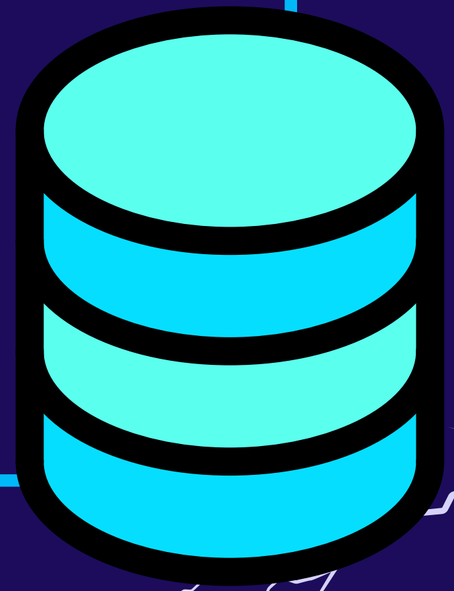
Assma Fadhli
@SpectrumShift DevOps Odyssey

PERSISTENT VOLUMES (PVS)

Persistent Volumes provide storage abstraction for use in Kubernetes.

- Cluster resources that administrators set up for persistent data storage.
- Independent of pod lifecycle—data remains available across pod restarts.
- Multiple types of PVs exist (NFS, iSCSI, cloud providers).

Read More >>



Assma Fadhli
@SpectrumShift DevOps Odyssey

PERSISTENT VOLUME CLAIMS (PVCS)

Persistent Volume Claims are requests for storage by users.

- Define the size, access modes, and storage class required.
- Bind dynamically to available PVs in the cluster.
- Decouple storage provisioning from pods.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

STORAGE CLASSES

Storage Classes provide a way to describe the “classes” of storage offered.

- Enable dynamic provisioning of persistent volumes.
- Allow administrators to define quality-of-service levels (e.g., performance, backup).
- Used in conjunction with PVCs to obtain resources automatically.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CONFIGMAPS

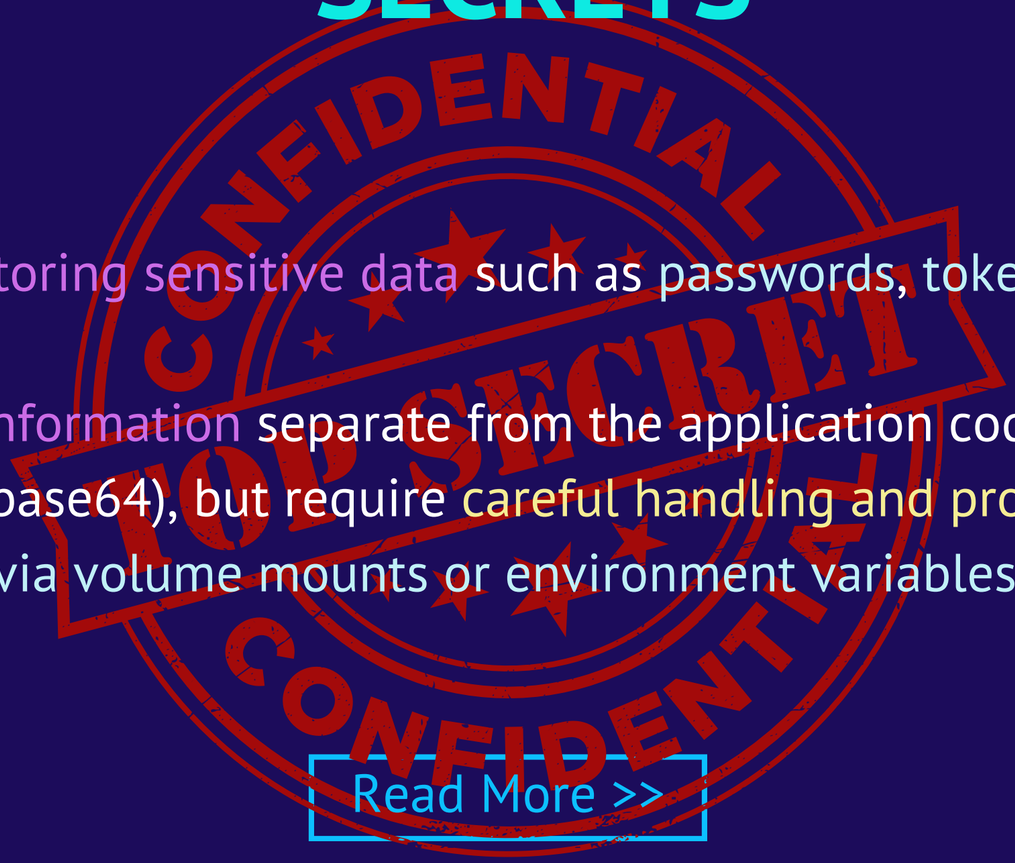
ConfigMaps store non-sensitive configuration data for applications.

- Decouple configuration artifacts from container images.
- Provide environment variables, command-line arguments, or configuration files to pods.
- Easily updated without rebuilding container images.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

SECRETS



Secrets are used for storing sensitive data such as passwords, tokens, or keys.

- Keep confidential information separate from the application code.
- Encoded (typically base64), but require careful handling and proper access control.
- Accessible to pods via volume mounts or environment variables.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

NAMESPACES

Namespaces allow partitioning of cluster resources between multiple users or teams.

- Provide scope for names, making it easier to manage environments.
- Help organize resources such as pods, services, and deployments.
- Isolate resources and improve security and resource management.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

LABELS

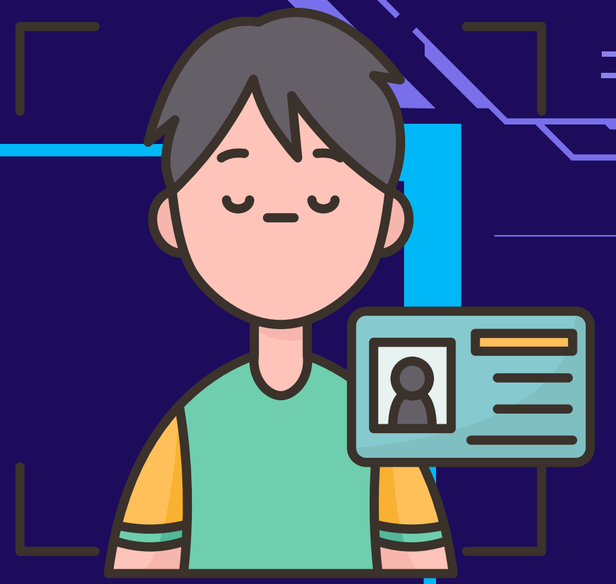
Labels are key/value pairs attached to resources.

- Facilitate grouping, selection, and management of Kubernetes objects.
- Useful for organizing and filtering components.
- Serve as the foundation for many Kubernetes operations like deployments and service selection.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

ANNOTATIONS



Annotations attach non-identifying metadata to Kubernetes objects.

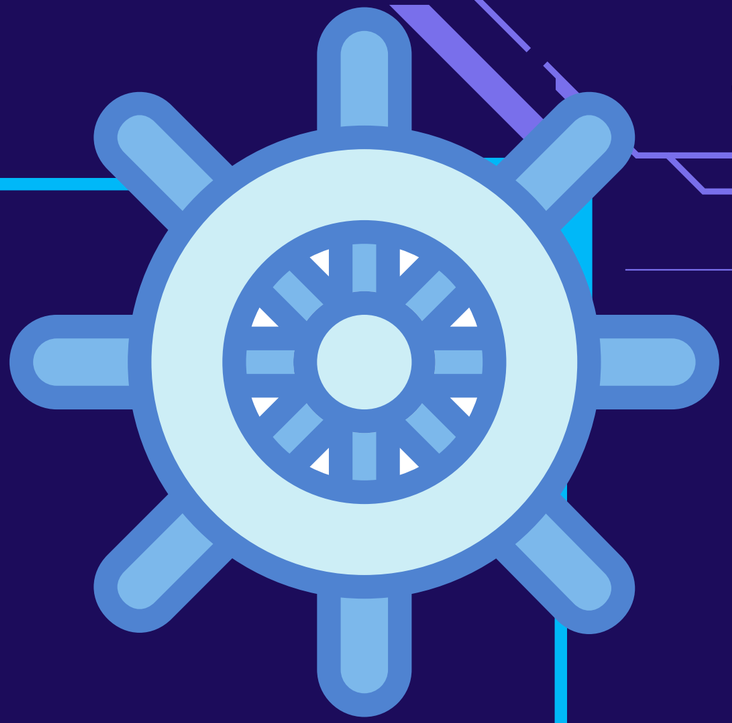
- Store arbitrary information that can be used by tools and libraries.
- Do not influence selection or grouping like labels do.
- Help provide context about configurations, automated processes, or debugging data.

Read More >>

Assma Fadhli

@SpectrumShift DevOps Odyssey

HELM



Helm is a package manager for Kubernetes.

- Simplifies the deployment and management of applications.
- Uses “charts” to define, install, and upgrade even the most complex Kubernetes applications.
- Manages templating, versioning, and dependency management.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

ROLE-BASED ACCESS CONTROL (RBAC)

RBAC manages authorization in Kubernetes.

- Assign permissions to users and groups based on roles.
- Controls who can access specific API objects and operations.
- Uses Roles (namespace-scoped) and ClusterRoles (cluster-wide) for fine-grained security.

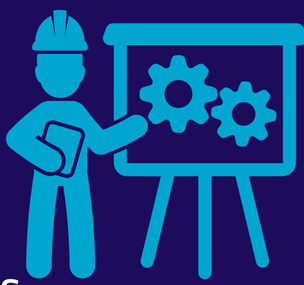
Read More >>

Assma Fadhli

@SpectrumShift DevOps Odyssey

KUBERNETES OPERATORS

Operators extend Kubernetes functionality by automating the management of complex applications.



- Encode operational knowledge into custom software.
- Manage stateful applications, backups, scaling, and other lifecycle events.
- Leverage Custom Resource Definitions (CRDs) to integrate with Kubernetes.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

NETWORK POLICIES

Network Policies specify how groups of pods are allowed to communicate.

- Provide firewall-like rules for pod interactions.
- Define ingress and egress rules using selectors for pods and namespaces.
- Enhance security by restricting unwanted traffic.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

SERVICE MESH OVERVIEW

Service meshes (e.g., Istio, Linkerd) extend Kubernetes networking.

- Provide advanced traffic management, security, and observability.
- Manage service-to-service communications with features like load balancing, retries, and circuit breaking.
- Operate alongside Kubernetes to enhance microservices architectures.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CLUSTER SETUP AND INSTALLATION

There are several ways to set up a Kubernetes cluster.

- Manual installation via kubeadm or kops.
- Managed services from cloud providers (GKE, EKS, AKS).
- Local development using Minikube or Kind.

Consider security, scalability, and network configuration during setup.

Read More >>

Assma Fadhli

@SpectrumShift DevOps Odyssey

MINIKUBE FOR LOCAL DEVELOPMENT

Minikube is a tool **for running a single-node Kubernetes cluster locally**.

- Ideal for learning, experimentation, and development.
- Provides a simple setup to run Kubernetes on your laptop.
- Supports various container runtimes and Kubernetes versions.

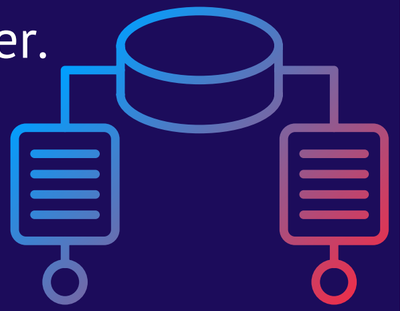
Read More >>

Assma Fadhli

@SpectrumShift DevOps Odyssey

KUBEADM

kubeadm simplifies the process of bootstrapping a new Kubernetes cluster.



- Provides commands to initialize the control plane and join nodes.
- Handles certificates, tokens, and configuration details automatically.
- Recommended for production-grade installations with expert guidance.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

ADVANCED SCHEDULING

Kubernetes supports advanced scheduling features to optimize resource allocation.

- Custom scheduling policies can be defined for workload placement.
- Consider factors like resource requests/limits, affinity, taints, and tolerations.
- Ensures workloads run efficiently across cluster nodes.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

TAINTS AND TOLERATIONS

Taints allow nodes to repel certain pods, while tolerations let pods schedule onto nodes with matching taints.

- Used to prevent resource contention or isolate specific workloads.
- Provide precise control over which pods can run on which nodes.
- Enhance cluster stability and performance.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

NODE AFFINITY AND ANTI-AFFINITY

Node Affinity allows you to constrain which nodes your pod can be scheduled on based on labels.

- Positive affinity rules select preferred nodes.
- Anti-affinity rules prevent pods from being co-located when not desired.
- Improve workload distribution and fault tolerance.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

MONITORING IN KUBERNETES

Monitoring is essential for maintaining cluster health and performance.

- Tools such as Prometheus and Grafana are commonly used.
- Monitor metrics such as CPU, memory, network, and custom application stats.
- Enable proactive alerts and capacity planning.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

LOGGING IN KUBERNETES

Effective logging helps in troubleshooting and understanding system behavior.

- Centralized logging solutions (e.g., Fluentd, Elasticsearch, Kibana) gather log data from multiple nodes.
- Access logs from pods, nodes, and the control plane.
- Use log aggregation and indexing to streamline debugging efforts.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

TROUBLESHOOTING KUBERNETES

Troubleshooting involves identifying issues in resource deployment, network, or performance.

- Begin by checking pod status and events via kubectl.
- Review logs from kubelet, API Server, or application-specific logs.
- Use built-in diagnostic tools and dashboard solutions to help locate issues.

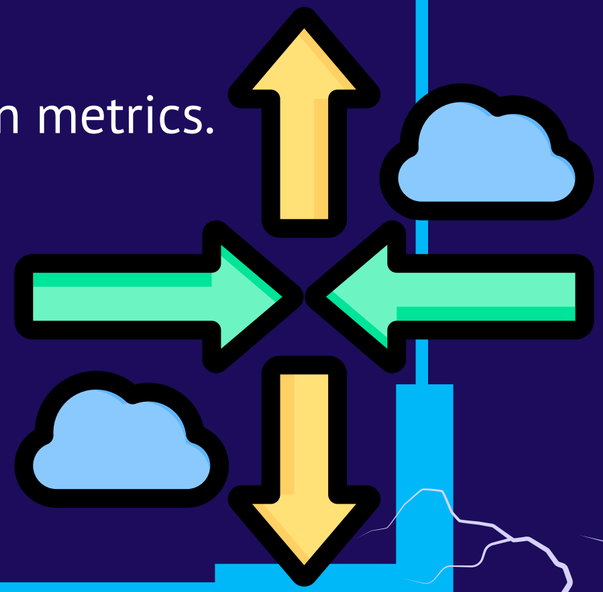
Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

SCALABILITY AND AUTO-SCALING

Kubernetes supports both manual and automatic scaling of applications.

- Horizontal Pod Autoscaler adjusts the number of pod replicas based on metrics.
- Cluster Autoscaler adds or removes nodes based on overall load.
- Design applications to be stateless and scalable for best results.



Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

UPGRADING A KUBERNETES CLUSTER

Kubernetes supports rolling upgrades to ensure minimal downtime.

- Upgrade sequentially starting with the control plane and then nodes.
- Use kubeadm or managed services to simplify the upgrade process.
- Carefully review release notes and test upgrades in staging environments.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

BEST PRACTICES AND SECURITY

Establish best practices to secure and efficiently manage Kubernetes clusters.

- Use RBAC, Network Policies, and Secrets management for security.
- Regularly update the cluster and its components.
- Implement resource requests/limits and monitor cluster health continuously.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

KUBERNETES APIS AND CUSTOM RESOURCES (CRDS)

Kubernetes exposes a rich API to interact with its components.

- Custom Resource Definitions (CRDs) allow you to extend Kubernetes for custom needs.
- Define, store, and manage domain-specific objects using CRDs.
- Integrate seamlessly with standard Kubernetes tooling and controllers.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

WRITING CUSTOM OPERATORS

Custom Operators encode operational knowledge in software.

- Leverage CRDs to define application-specific resources.
- Operators automate tasks such as deployment, scaling, and backups.
- Use Operator Frameworks and SDKs to simplify development and management.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

CASE STUDIES AND REAL-WORLD EXAMPLES

Successful Kubernetes deployments come from a range of industries.

- Learn from case studies of companies that migrated to Kubernetes.
- Understand challenges faced during deployment, scaling, and maintenance.
- Explore real-world examples of microservices, CI/CD pipelines, and automated operations.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

THE FUTURE OF KUBERNETES

Kubernetes continues to evolve as container orchestration needs expand.

- Integration with emerging technologies like serverless and edge computing is underway.
- Growing ecosystem and community support encourage innovation.
- Stay updated with new releases, features, and best practices.

Read More >>

Assma Fadhli
@SpectrumShift DevOps Odyssey

THIS GUIDE

Provides a comprehensive overview of Kubernetes fundamentals and advanced topics.

- Reviewed architecture, core components, scheduling, and network concepts.
- Covered storage, security, and management best practices.
- Prepared you for further exploration and real-world deployment of Kubernetes systems.

Assma Fadhli
@SpectrumShift DevOps Odyssey

FOUND THIS HELPFUL? SHARE IT
AND FOLLOW FOR MORE!

Assma Fadhli

@SpectrumShift DevOps Odyssey