

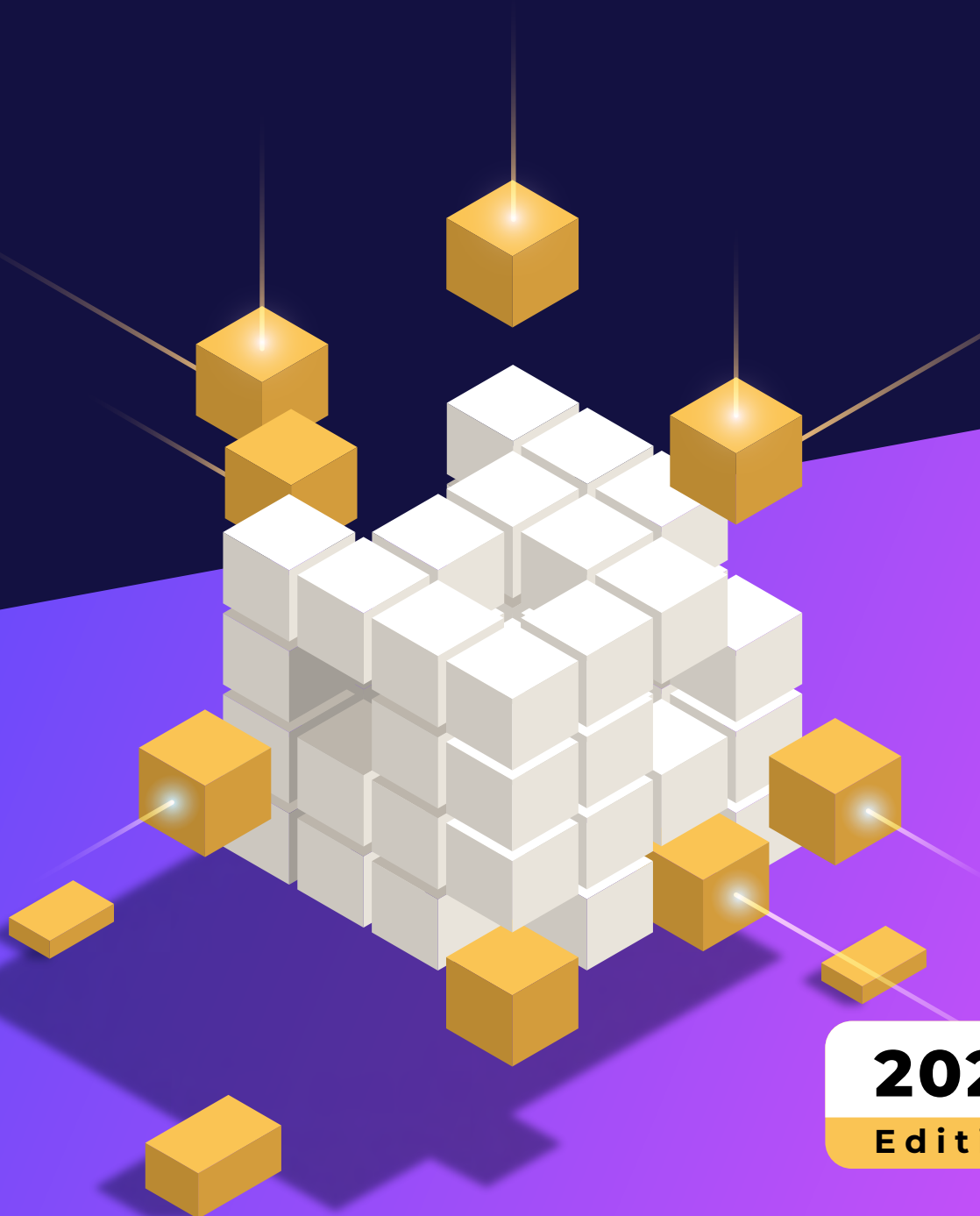
GlobalDots

Cloud Innovation Hunters

DevOps Practical Guide

Efficient Cluster Management

K8s Best Practices



2024

Edition

Table of Contents

Executive Summary	3
Section 1: Managed Kubernetes Services.....	4
Section 2: Resource Management and Optimization.....	5
Section 3: Monitoring and Logging.....	7
Section 4: Implement GitOps for Kubernetes.....	8
Section 5: Best Practices on EKS, GKE, and AKS.....	9
About GlobalDots	11

Executive Summary

Kubernetes is a powerful container orchestration platform that can help developers and DevOps teams deploy, manage, and scale applications more efficiently. However, with great power comes great responsibility. It is known that Kubernetes clusters can become complex and challenging, stretching beyond human capacity for efficient management.

In this e-book, we will discuss some best practices for efficient Kubernetes cluster management for developers and DevOps teams. We will also cover some specific best practices for managing Kubernetes clusters on EKS, GKE, and AKS.

Why listen to us?

At GlobalDots, we pride ourselves on being dedicated innovation hunters with a keen eye for groundbreaking cloud solutions. Having over 20 years of experience in the field, and serving more than 350 clients globally, our expertise lies not just in identifying, but also in providing a swift and efficient pathway to cloud innovation. We specialize in offering strategies and curated solutions that are not only innovative but also practical, focusing on reducing costs and saving you time.

Our approach is tailored to your needs, making sure that the solutions we provide align directly with your organizational objectives. With two decades of experience, we have honed our ability to foresee and adapt to the evolving landscape of cloud technology, offering you the most relevant and effective solutions.

Trusted by



Lufthansa

similarweb



IBM



amdocs

Payoneer

SentinelOne

Playtika

eToro

BOSCH



NATIVE INSTRUMENTS

mobileye

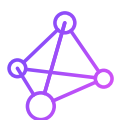
AppsFlyer

GONG

Section 1: Managed Kubernetes Services

Use a Managed Kubernetes Service

Managed Kubernetes services such as EKS, GKE, and AKS take care of the heavy lifting of managing your Kubernetes cluster, such as:



Provisioning and configuring nodes:

Managed Kubernetes services automatically provision and configure nodes for you, so you don't have to worry about the underlying infrastructure.



Handling upgrades:

Managed Kubernetes services handle upgrades to your cluster automatically, so you don't have to worry about downtime or compatibility issues.



Providing security features:

Managed Kubernetes services provide a variety of security features, such as encryption, access control, and auditing, to protect your cluster and applications. Using a managed Kubernetes service can free up your time to focus on other tasks, such as developing and deploying applications.



Section 2: Resource Management and Optimization

Implement Resource Management

Resource management is important for ensuring that your applications have the resources they need to run efficiently. Kubernetes provides features such as resource quotas and limits to control the amount of resources that each application can use.



Resource Quotas:

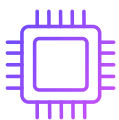
Specify the maximum amount of resources that an application can use. This can help to prevent applications from consuming too many resources and causing performance problems for other applications.



Resource Limits:

Specify the maximum amount of resources that an application can use at any given time. This can help to prevent applications from hogging resources and causing performance problems for the entire cluster.

Resource Management Tips



Instance Types:

When choosing instance types for your Kubernetes cluster, consider CPU, memory, and storage requirements, as well as cost-effectiveness. Cloud providers offer a variety of instance types to match your cluster's needs.



Nodes Autoscaling:

Autoscaling solutions like Cluster Autoscaler and Karpenter are popular for Kubernetes due to their ability to dynamically adjust your cluster's capacity based on demand. By integrating and configuring these solutions, you can ensure that your infrastructure scales effortlessly, benefiting from automatic resource provisioning and de-provisioning. This not only optimizes performance by preventing resource shortages during traffic spikes but also minimizes operational overhead and reduces costs during periods of low usage.

Pods Autoscaling:

Leverage Horizontal Pod Autoscaling (HPA):



Horizontal Pod Autoscaling allows your applications to dynamically adjust the number of replicas based on resource utilization, ensuring efficient resource allocation and optimal performance as workloads fluctuate.

Utilize Vertical Pod Autoscaling (VPA):

VPA dynamically adjusts resource requests and limits for pods based on their usage history, optimizing resource allocation and improving application

Quality of Service (QoS) Classes:



Quality of Service classes (Guaranteed, Burstable, BestEffort) allow you to define how pods should be prioritized and scheduled. This can help in ensuring that critical workloads receive the resources they need, while less critical workloads gracefully degrade when resources are scarce.

Optimize Storage Provisioning:



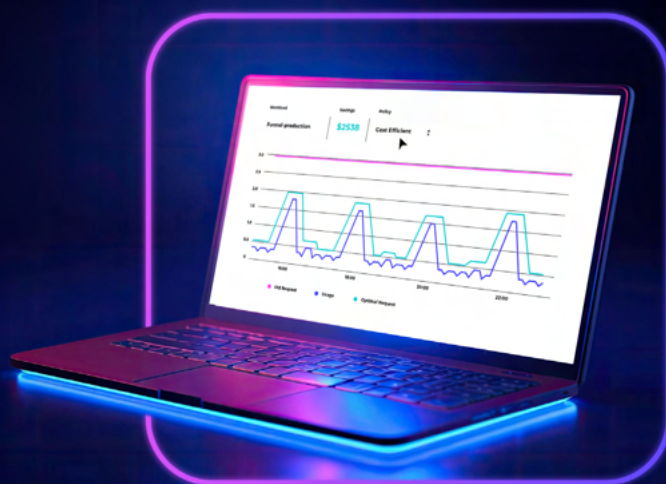
Efficient storage management is crucial. Consider implementing dynamic storage provisioning to allocate storage resources only when needed, and use PersistentVolumeClaims (PVCs) efficiently to avoid over-provisioning.

Minimizing Container image Size



Reducing the size of your container images offers several advantages. It accelerates build and deployment processes and decreases resource consumption on your Kubernetes (K8s) cluster. To achieve this, consider eliminating unnecessary packages and prioritize using compact OS distribution images like Alpine. Smaller images not only load faster but also occupy less storage space.

Additionally, this practice enhances security by reducing potential attack vectors, making it more challenging for malicious actors to exploit vulnerabilities within your containers.



Section 3: Monitoring and Logging

Use Monitoring and Logging

Monitoring and logging are essential for tracking the health and performance of your Kubernetes cluster and applications. Kubernetes provides a variety of tools for monitoring



The Kubernetes Dashboard:

A graphical user interface that provides a real-time view of your cluster and application performance.



The kubectl CLI tool:

A command-line tool for monitoring and managing your Kubernetes cluster and applications.



Prometheus:

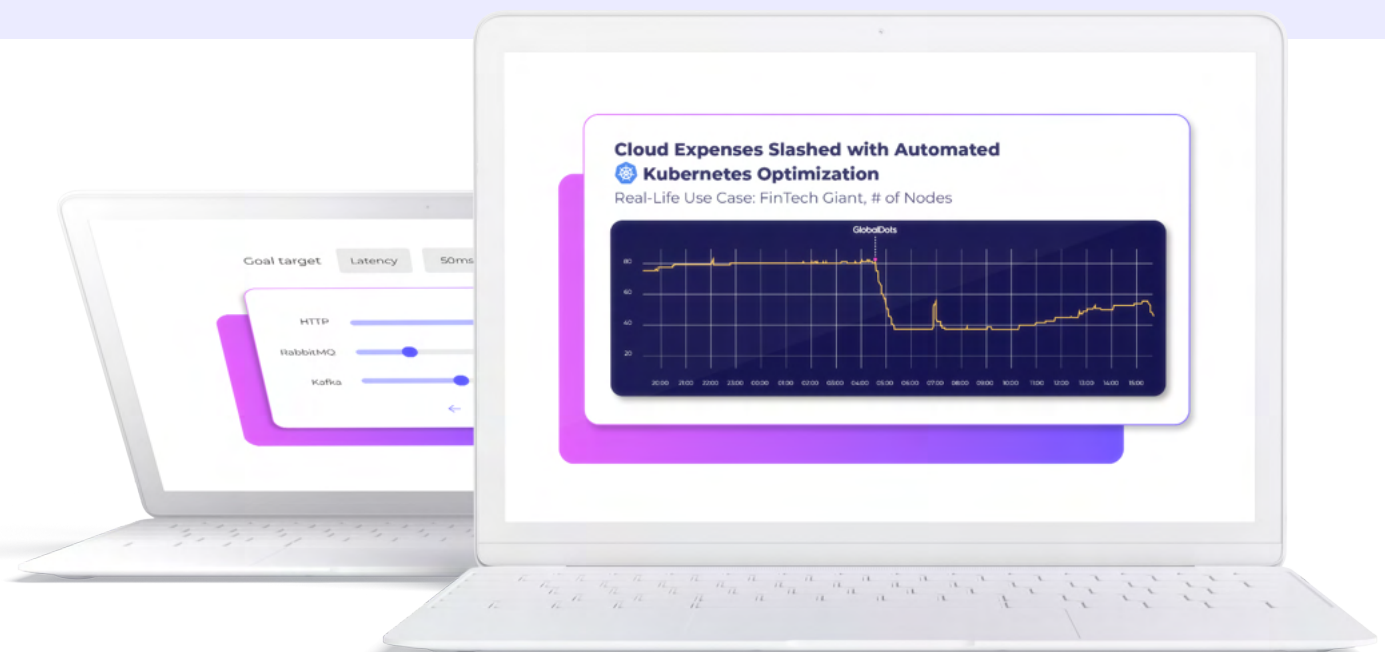
A monitoring system that collects metrics from your cluster and applications.



Elasticsearch:

A logging system for storing and searching logs from your cluster and applications.

By using monitoring and logging tools, you can quickly identify and troubleshoot problems with your cluster and applications.



Section 4: Implement GitOps for Kubernetes

Implement Resource Management

GitOps is a declarative and robust approach to managing Kubernetes configurations, where Git serves as the single source of truth. With GitOps, the state of your Kubernetes cluster is continuously synchronized with a Git repository, ensuring that the desired state is always maintained. This method offers numerous advantages, including improved security, version control, auditing, and compliance.

Examples of GitOps Tools

ArgoCD:



ArgoCD is a popular GitOps tool that provides a user-friendly interface for managing Kubernetes applications. It monitors a Git repository for changes to application definitions and automatically deploys and syncs applications with the desired state. ArgoCD ensures that your cluster is always aligned with your Git repository, simplifying application deployment and management as the ones GlobalDots curate.

Flux



Flux is another widely used GitOps tool that automates the deployment and scaling of applications in your Kubernetes cluster. It continuously updates the cluster to match the versions defined in the Git repository, making it a powerful choice for maintaining consistency and reliability.



Section 5: Best Practices on EKS, GKE, and AKS

Best Practices for Managing Kubernetes Clusters on EKS

Amazon EKS Autoscaling solutions:

Karpenter

Karpenter is a high-performance Kubernetes cluster autoscaler that enhances application availability and cluster efficiency.



It swiftly deploys appropriately sized compute resources, such as Amazon EC2 instances, in response to shifting application demands.

By integrating with Kubernetes and AWS, Karpenter efficiently allocates resources tailored to workload requirements, encompassing compute, storage, acceleration, and scheduling needs. For additional details, refer to the Karpenter documentation.

EKS Cluster Autoscaler



The Cluster Autoscaler automatically adjusts the number of nodes in your cluster when pods fail or are rescheduled onto other nodes. The Cluster Autoscaler uses Auto Scaling groups. For more information, see Cluster Autoscaler on AWS.

Use Amazon EKS Managed Node Groups:



Managed node groups automate node provisioning and management, which can save you time and effort.

Use EKS Blue/Green deployments:



Reduce downtime and minimize risk when deploying new applications. Amazon EKS Blue/Green deployments allow you to create a new environment (the “blue” environment) alongside the existing one. Once the new application is healthy, you can seamlessly switch traffic to the blue environment, ensuring minimal disruption to your users.

Use EKS Fargate



Consider using Amazon EKS Fargate, a serverless compute environment for running your Kubernetes applications. With Fargate, you can abstract away the management of nodes entirely, allowing you to focus solely on your applications. This can be especially useful for applications with variable workloads.

Best Practices for Managing Kubernetes Clusters on GKE



Use Google Kubernetes Engine (GKE) Autopilot:

Simplify cluster management with automation for provisioning, upgrades, and security features.



Leverage GKE Workspaces:

Provide developers with self-service sandbox environments for development and testing.



Employ GKE Node Auto Scaling:

Automatically scale the number of nodes based on demand for improved performance and cost-effectiveness.

Best Practices for Managing Kubernetes Clusters on AKS



Utilize Azure Kubernetes Service (AKS) Managed Node Groups:

Simplify node management with automatic provisioning and management.



Explore AKS Virtual Kubelet:

Run containerized applications on Azure virtual machines (VMs) without deploying a Kubernetes cluster on each VM.



Enable AKS cluster autoscaler:

Automatically scale the number of nodes based on demand for improved performance and cost-effectiveness.

By following these best practices, you'll not only streamline the management of your Kubernetes clusters but also unlock the true agility and efficiency that Kubernetes promises, without the operational complexities overhead.

About GlobalDots

With 20 years of experience in Cloud Innovation, Security, Web Performance, DevOps, Corporate IT, and advanced AI/ML models - GlobalDots now grants businesses privileged access to groundbreaking FinOps technologies & services.

Our mission is to help all organizations close knowledge gaps and manage their spend prudently. Whether you are a startup or an established enterprise, we'll customize our solutions to meet the unique needs of your IT architecture. We leverage our vast experience from hundreds of successful use cases to ensure optimal results.

