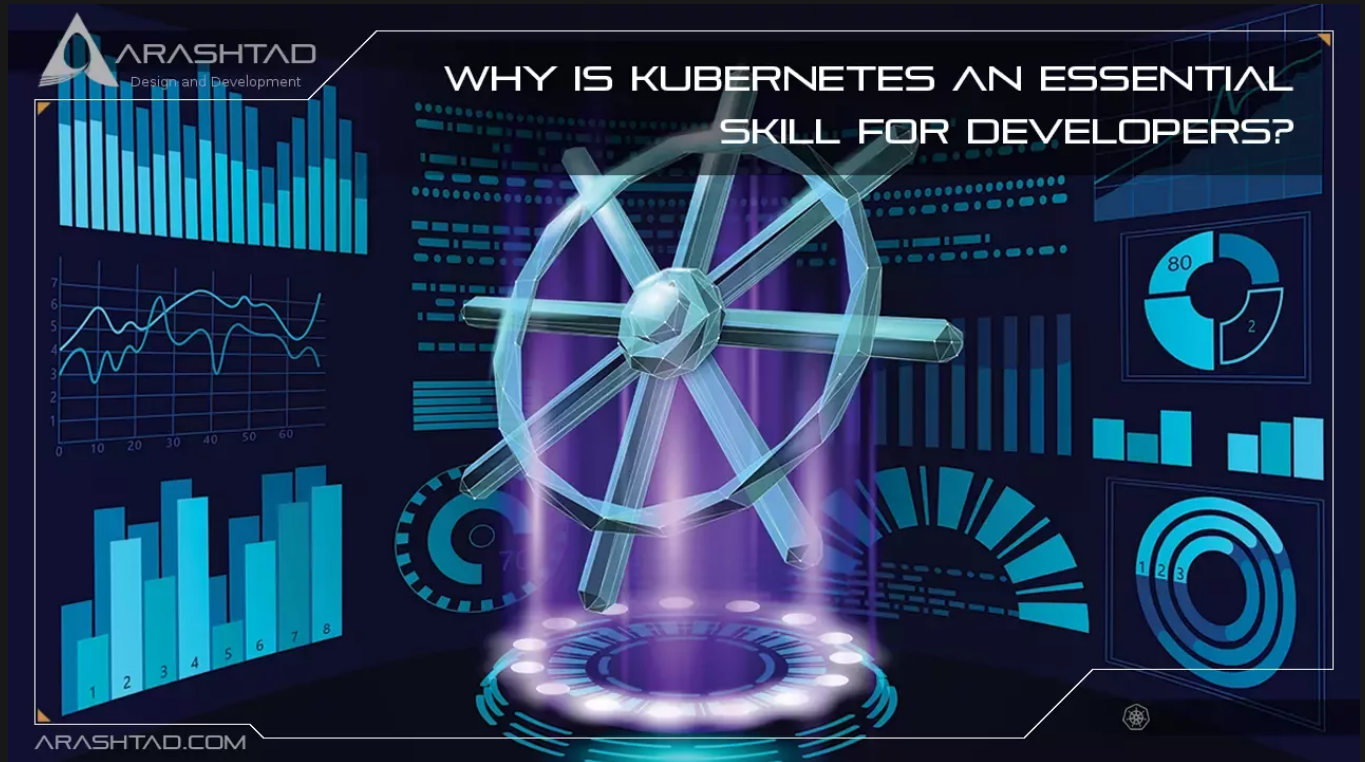


Why Is Kubernetes an essential skill for Developers?

No comments



Kubernetes is an open-source container orchestration platform that allows you to automate running and orchestrating container workloads. It is a powerful tool that offers a huge ecosystem of tools — package managers, service meshes, source plugins, monitoring tools, and more — as an abstraction layer for deploying standardized, full-stack applications across an ever-increasing range of platforms. K8s is often referred to as “K8s” Kubernetes is not going away, at least not anytime soon. And, increasingly, developers are being required to interact with it.

Kubernetes Overview

Although DevOps has revolutionized the world of IT, there are some flaws in the 'DevOps plan' that could affect your workflows drastically. To counter these issues in DevOps, several companies have come up with tools and frameworks that enhance your workflows by making them more efficient and effective. One of these tools is Kubernetes, a container management utility developed by Google.

For those of us who don't know K8s, it's a container management or orchestration system that can be used to automatically deploy and scale containers. K8s complements existing tools such as Docker and Rocket that are used in DevOps for container creation. K8s is now an essential skill to bag a DevOps job. Now that we know what it is and what it does, let's examine why it is now so popular.

Why Do We Need Kubernetes?

Project and operations management is becoming increasingly difficult in today's IT world due to its increasing scale and complexity. K8s counteracts this by automatically scaling and managing containers that are used to deploy and test project or application modules. The tool is also extremely easy to use, learn, and integrate with your existing projects.

Industry Job Trends

Taking up a Kubernetes course, such as that offered by Edureka, is another reason to consider it. Several job postings for DevOps professionals now require you to understand container orchestration with K8s or Docker Swarm, based on job postings on leading job portals like TimesJobs. There are several openings in technology giants like Microsoft, IBM, and Infosys for DevOps experts with Kubernetes expertise. There are also hundreds, if not thousands, of startups that use K8s exclusively for their business. As a result, learning K8s will not only benefit experienced professionals, but even entry-level professionals can find lucrative jobs by learning K8s.

Kubernetes Salary Trends

In the USA, the average salary for a DevOps K8s engineer is \$145,525 per year, which is \$69.96 per hour. Approximately \$130,000 is the starting salary for entry-level jobs, while \$177,500 is the average salary for most experienced workers.

Kubernetes is one of those promising technologies that can boost your career prospects in the years to come. Therefore, if you are looking for a dynamic job with a large salary, then adding K8s to your technology portfolio would be the best move you can make.

What Developers and Operations Teams Need to Know

There are a few things that developers and operations engineers need to know about what their peers do.

1. To make an informed decision, they must understand the specific characteristics of their chosen cloud provider in comparison to other providers. This knowledge should apply whether the cloud is public, private, or hybrid.
2. When it comes to resourcing their applications, they need to be aware of the financial impact and know how to reduce costs and eliminate waste. In the cloud, it's easy to set up a new environment and infrastructure, so it's easy to

overlook how quickly those costs can add up if we mismanage them. Considering auto-scaling policies and how they affect costs is a good idea, for example.

3. It is imperative that they have knowledge of application performance management, especially the tools and techniques that are used to analyze and improve application performance.
4. When an incident occurs, they need to know how to deal with it appropriately and escalate it when needed. DevOps is fundamentally based on accepting failure and finding ways to mitigate it, so handling incidents effectively and efficiently when they occur is crucial.

So that all teams are aware of any shortcomings in their tools and applications and how the developers might resolve them, they need to establish feedback loops on both sides of the development fence. sharing ownership of tools and environments is a great way to accomplish this.

Why is Kubernetes an essential skill for Developers?

What are the specific Kubernetes skills that developers should learn?

1. From code check-in to building, testing, and deployment, and how their organization's CI/CD system works.
2. Pods and containers in K8s.
3. Interactions between K8s networking and applications, including services, DNS, and load balancing.
4. Minikube, kubectl, helm, kind, and the K8sdashboard are commonly used tools for local testing and modeling their deployment.
5. Debugging clusters and containers when things go wrong by monitoring, logging, and debugging.
6. However, these are a great starting point for both operations and development teams to understand how they can work together.

Shifting Left

Additionally, there is a lot of discussion in the software development world about "shifting left," along with DevOps and SRE adoption. Shifting left involves focusing on early detection and prevention of problems in the software development lifecycle (SDLC) to improve quality. CI/CD pipelines, testing practices, and robust, automated continuous integration/continuous delivery (CI/CD) pipelines are examples of how this works. The shift to the left applies to operational practices as well. In the past, developers coded their applications and handed them off to operations for

deployment. Since then, things have changed dramatically, and old models are no longer effective.

Conclusion

An application's platform must be understood. successful engineering organizations work diligently to ensure development and operations teams do not work in silos. By collaborating early in the software development lifecycle, they aim to ensure that all teams involved in the process have a clear understanding of coding, building, testing, and deployment. Kubernetes isn't the only skill developers need to master, but they should also be proficient in other skills that can affect the performance of their applications. The application puzzle is incomplete without CI/CD, deployments to production, monitoring, and understanding CPU, memory, cluster and pod health.

Follow Arashtad on Social Media

RSS FEED