Kubernetes Hosting:

"Kubernetes is a portable, extensible, open source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation."

To begin my essay on Kubernetes Hosts it is only natural to start with local hosting. Local hosting is an easy way of getting into Kubernetes as it lets you try it out without having to pay anything. Furthermore it lets you set it up as a local development environment, making the separation between development- and production environment simple. Now to look at a specific example. "Minikube" is one of the most popular choices to run Kubernetes locally. It is doesnt require a specific operating system and is really simple to install. It furthermore offers all Kubernetes features available to local environments and gives a direct API endpoint. Looking into the tradeoffs reveals a couple weaknesses. With running any application locally comes the drawback of only having the local resources to work with, limits the scalability and complexity of said application. The same is true for Minikube, thus it is not suitable for production-scale applications. But that can also be a benefit. When looking into the security aspect, we see that running Minikube is less of a concern, because it is intended for development. Furthermore there is no associated cost with running Minikube. Though there are some maintenance implications. It requires regular updates by the user, due to it being open source. Now coming to the cluster creation of Minikube. It creates a single-node cluster. Going onto the the CI/CD integration. Minikube can easily be connected into a CI/CD workflow through systems like Github Actions, and can be used for building and testing a system.

Continuing the discussion, it's essential to explore managed hosting. I focused on google kubernetes engine GKE for this aspect. As a managed hosting solution, GKE represents a stark contrast to local hosting methods like Minikube. Managed by Google Cloud, GKE offers a highly scalable and secure environment for Kubernetes deployments, eliminating much of the manual configuration and maintenance required in local setups. This is perfect for production-scale applications where reliability, availability, and scalability are crucial. GKE integrates easily with Google Cloud's infrastructure, providing users with access to a wide range of cloud-based resources and services. This integration also simplifies many operational tasks, such as load balancing, node provisioning, and automated scaling, therefore allowing developers to focus more on application development than infrastructure management. Though, this convenience comes at a costl, as users must pay for the hosting and maintenance of the deployment. Despite these trade-offs, GKE's robustness and ease of use make it an attractive option for enterprises looking to leverage Kubernetes in a more hands-off, cloud-centric environment.

To now come to the last aspect im going to look at self-managed hosting through manual or automated bootstrapping. It offers a balance between control and ease of use. Manual bootstrapping provides complete control over Kubernetes clusters, ideal for organizations with specific needs, but demands a deep understanding of Kubernetes and significant human resources. Automated bootstrapping, with tools like Kubeadm, simplifies the setup while retaining customization flexibility. This method reduces complexity and potential errors while offering more flexibility than fully managed solutions like GKE. A drawback is that it requires organizations to manage and maintain their infrastructure, including updates and security. All that makes it suitable for anyone with the expertise and a need for a custom Kubernetes environment.

To now come to a conclusion. All the previously mentioned hosting options are valid for their target audience. Minkube for anyone starting out with kubernetes and or still experimenting. A managed solution for a production environment that has to run on a cloud system, and needs the security offered. And a self-managed variant for anyone that has the needed knowhow of kubernetes and works on a niche project that doesn't fit into the normal hosting methods. With all that in mind I would choose Minikube or a similar local hosting method. It offers everything I need as a person that is interested in trying out kubernetes but doesn't require it on a cloud level. Furthermore I don't have the expertise needed to work with a self-managed variant.

Sources:

-https://kubernetes.io/docs/concepts/overview/
-https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-intro/
-https://medium.com/avmconsulting-blog/ci-cd-with-kubernetes-minikube-6e9e0b2150b2
-https://www.airplane.dev/blog/managed-kubernetes-services
-https://packetpushers.net/should-you-choose-a-manged-kubernetes-service-or-self-host-kubernetes-in-the-cloud/