

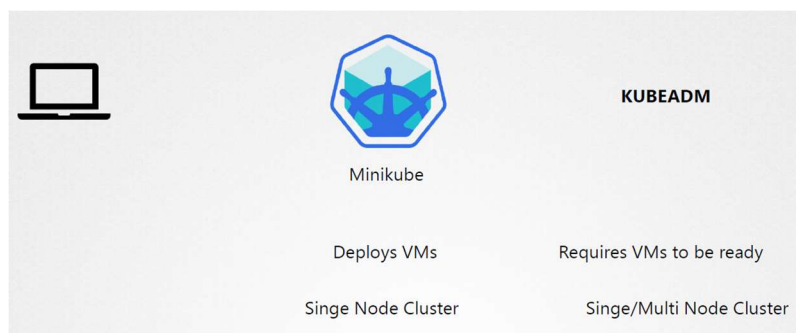
Choosing Kubernetes Infrastructure

In this lecture, we talk about the different choices available for the infrastructure hosting a Kubernetes cluster. In the previous lecture, we discussed the various options we have in deploying a Kubernetes cluster. Let's look at it in a bit more detail here. Kubernetes can be deployed on various systems in different ways, starting with your laptops to physical or virtual servers within your organization as well as those in the cloud. Depending on your requirements, your cloud ecosystem, and the kind of applications you wish to deploy, you may choose one of these solutions.

On a laptop or a local machine, there are a number of ways to get started. First of all, on a supported Linux machine, you can get started with installing the binaries manually and setting up a local cluster. However, that is too tedious, especially if you are just getting started. So relying on a solution that automates all that will help in setting up a cluster in a matter of minutes. We will look at some of those solutions in a bit.

On Windows, on the other hand, you cannot set up Kubernetes natively as there are no Windows binaries. You must rely on a virtualization software like Hyper-V or VMware Workstation or VirtualBox to create Linux VMs on which you can run Kubernetes. There are also solutions available to run Kubernetes components as Docker containers on Windows VMs. But remember, even then, the Docker images are Linux-based, and under the hoods, they're run on a small Linux OS created by HyperV for running Linux Docker containers.

So what are some of the solutions available to easily get started with Kubernetes on a local machine? Minikube deploys a single node cluster easily. It relies on one of the virtualization software like Oracle VirtualBox to create virtual machines that run the Kubernetes cluster components. We have seen this in the beginner's course. The Kubectl tool can be used to deploy a single node or a multi-node cluster real quick. But for this, you must provision the required host with supported configuration yourself. So the difference between the first two and Kubectl is that the first two provisions a VM with supported configuration by itself. Whereas Kubectl expects the VMs provisioned already. At the same time, it allows for deploying multi-node clusters, whereas the former doesn't. Again, deploying a Kubernetes cluster locally on a laptop is usually for learning, testing, and development purposes.



For production purposes, there are many ways to get started with a Kubernetes cluster, both in a private or a public cloud environment. I would categorize them as turnkey solutions, or hosted or managed solutions. Turnkey solutions are where you provision the required VMs, and use some kind of tools or scripts to configure Kubernetes cluster on them. At the end of the day, you are responsible for maintaining those VMs and patching them and upgrading them, et cetera. But cluster management and maintenance

are mostly made easy using these tools and scripts. For example, deploying a Kubernetes cluster on AWS using the KOPS tool.

Hosted solutions are more like Kubernetes as a service solution, where the cluster along with the required VMs are deployed by the provider, and Kubernetes is configured by them by the provider. The VMs are maintained by the provider, for example, Google Container Engine lets you deploy a Kubernetes cluster in a matter of minutes without you having to perform any configuration by yourself.

Turnkey Solutions	Hosted Solutions (Managed Solutions)
<ul style="list-style-type: none">• You Provision VMs• You Configure VMs• You Use Scripts to Deploy Cluster• You Maintain VMs yourself• Eg: Kubernetes on AWS using KOPS	<ul style="list-style-type: none">• Kubernetes-As-A-Service• Provider provisions VMs• Provider installs Kubernetes• Provider maintains VMs• Eg: Google Container Engine (GKE)

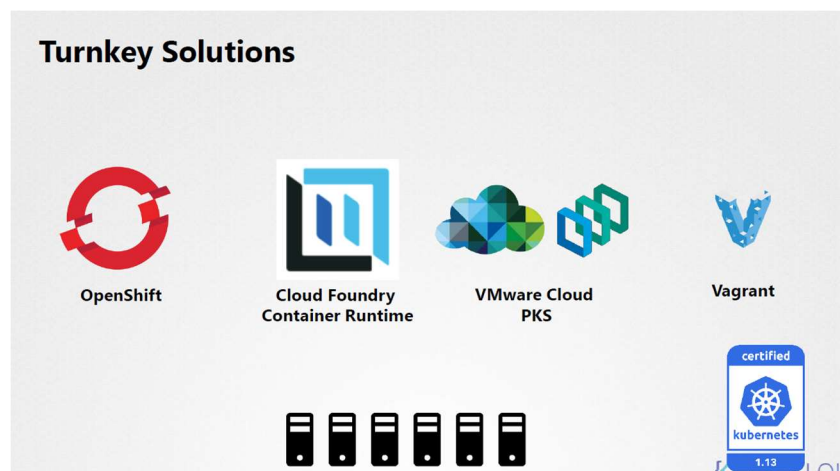
Let us look at some of the turnkey solutions.

OpenShift is a popular on-prem Kubernetes platform by Red Hat. For those of you who may not be familiar, OpenShift is an open source container application platform and is built on top of Kubernetes. It provides a set of additional tools and a nice GUI to create and manage Kubernetes constructs and easily integrate with CI/CD pipelines, et cetera.

Cloud Foundry Container Runtime is an open source project from Cloud Foundry that helps in deploying and managing highly available Kubernetes clusters using their open source tool called BOSH. If you wish to leverage your existing VMware environment for Kubernetes, then the VMware Cloud PKS solution is one that should be evaluated.

Vagrant provides a set of useful scripts to deploy a Kubernetes cluster on different cloud service providers. All of these solutions make it easy to deploy and manage a Kubernetes cluster privately within your organization. You must have a few virtual machines with supported configurations in place.

These are a few of the many Kubernetes certified solutions. While there are many more, so check them out in the Kubernetes documentation page.



Let us look at some of the hosted solutions. Google Container Engine is a very popular Kubernetes as a service offering on Google Cloud platform. OpenShift Online is an offering from Red Hat where you gain access to a fully functional Kubernetes cluster online. Azure has the Azure Kubernetes Service. And finally, Amazon Elastic Container Service for Kubernetes is Amazon's hosted Kubernetes offering. Again, these are just some of the solutions. There are many more.

Hosted Solutions



Google Container Engine (**GKE**)



OpenShift Online



Azure Kubernetes Service



Amazon Elastic Container Service for Kubernetes (EKS)